

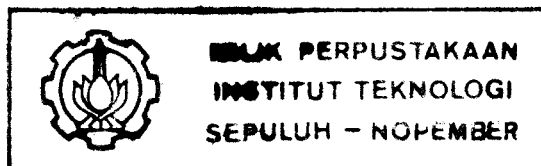
3100096007590

**PENERAPAN MATH COPROCESSOR 8087
PADA MAKSIMUM SISTEM 8088
SEBAGAI SARANA PRAKTIKUM**

PERPUSTAKAAN ITS	
Tgl. Terima	22 SEP 1994
Terima Dari	H
No. Agenda Prp.	2731



RSE
621.391 G
Wir
P-1
1994



OLEH :

IG.A.K. DARMA WIRATA

Nrp. 2882201029

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA**

1994

**PENERAPAN MATH COPROCESSOR 8087
PADA MAKSIMUM SISTEM 8088
SEBAGAI SARANA PRAKTIKUM**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik Elektro**

Pada

Bidang Studi Elektronika

Jurusan Teknik Elektro

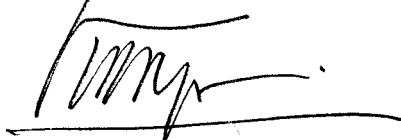
Fakultas Teknologi Industri

Institut Teknologi Sepuluh Nopember

Surabaya

Mengetahui / Menyetujui

Dosen Pembimbing



(Ir. Karyadi, Msc.)

SURABAYA

Agustus, 1994

Dengan ilmu

hidup menjadi mudah

Dengan Agama

hidup menjadi terarah dan bermakna

Kupersembahkan untuk :

Bapak dan Ibu tercinta,

mas Krisna, mbak Laksmi, mas Suta dan dik Putu tercinta

ABSTRAK

Operasi perhitungan dengan mikroprocessor 8088 dapat dilakukan dengan kecepatan dan keakuratan yang tinggi. Namun memiliki keterbatasan. Terutama pada proses memakai bilangan real. Untuk itu dibutuhkan prosessor tambahan yaitu Math coprocessor 8087. Dan untuk dapat bekerja bersama-sama dengan mikroprosesor tambahan ini maka mikroprosesor 8088 harus dioperasikan pada mode maksimum.

Pada tugas akhir ini akan dirancang suatu modul yang mengaplikasikan math coprocessor 8087 pada mode maksimum 8088. Dengan maksud sebagai sarana praktikum memprogram dan menerapkan 8087 untuk aplikasi yang sederhana. Sehingga dapat dipergunakan pada laboratorium bagi mahasiswa yang ingin mempelajari mikroprosesor 8087.

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas segala rahmat dan petunjuk-Nya sehingga penulis dapat merampungkan Tugas Akhir yang berjudul :

PENERAPAN MATH CO-PROCESSOR 8087

PADA MAKSIMUM SISTEM 8088 SEBAGAI SARANA PRAKTIKUM

Tugas Akhir ini mempunyai beban 6 SKS (Satuan Kredit Semester) yang disusun untuk memenuhi persyaratan menyelesaikan studi S-1 pada Jurusan Teknik Elektro, Fakultas Teknik Industri, Institut Teknologi Sepuluh Nopember Surabaya.

Akhirnya tidak lupa penulis mengucapkan banyak terima kasih kepada :

1. Bapak Ir. Karyadi, MSc selaku Dosen Pembimbing I yang telah memberikan bimbingan, petunjuk dan saran hingga selesainya Tugas Akhir ini.
2. Bapak Ir. Harris Pirngadi selaku Dosen Pembimbing II yang telah memberikan bimbingan dan saran hingga selesainya Tugas Akhir ini.
3. Bapak Ir. Soetikno selaku Koordinator Bidang Studi Elektronika.
4. Bapak-bapak Dosen, khususnya dosen pada bidang studi Elektronika yang telah memberikan bekal ilmu selama penulis menempuh kuliah.

5. Bapak dan Ibu tercinta, yang telah membimbing penulis sejak kecil dengan penuh kasih sayang, serta mas Krisna, mbak Laksmi, mas Suta dan dik Jaya tercinta.
6. Tersayang Iis yang dengan sabar dan kerap kali memberikan dorongan semangat kepada penulis hingga terselesainya Tugas Akhir ini.
7. Semua pihak yang telah membantu dalam penyelesaian Tugas Akhir ini.

Penulis berharap bahwa Tugas Akhir ini akan bermanfaat bagi para pembaca serta dapat lebih disempurnakan dan dikembangkan lebih lanjut.

Surabaya, Juli 1994

Penulis

DAFTAR ISI

	Hal
Lembar Judul	i
Lembar Persetujuan	ii
Lembar Persembahan	iii
ABSTRAK	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	xii
BAB I. PENDAHULUAN	1
I.1. Latar Belakang	1
I.2. Permasalahan	2
I.3. Pembatasan Masalah	2
I.4. Tujuan	3
I.5. Metodologi	3
I.6. Sistematika Pempahasan	4
I.7. Relevansi	6
BAB II. TEORI PENUNJANG	7
II.1. Tujuan Umum	7
II.1.1. Mode Minimum 8088	11
II.1.2. Mode Maximum μ P 8088	13
II.2. Math Co-Processor 8087	27
II.3. Konversi Analog Digital	39
II.3.1. Konversi Analog ke Digital	39
II.3.2. Konversi Digital ke Analag	41

II.3.2. Konversi Digital ke Analag	41
II.4. Kontrol Proses	44
II.4.1. Kontrol Analog	44
II.4.2. Kontrol Digital	52
BAB III. PERENCANAAN	55
III.1. Perencanaan Hardware	55
III.1.1. Rangkaian CPU, Clock dan Pembangkit Wait State	56
III.1.2. Rangkaian Pengontrol Bus	59
III.1.3. Rangkaian Buffer Bus	61
III.1.4. Rangkaian Interface Memory	63
III.1.5. Rangkaian Masukan dan Keluaran	65
III.1.6. Rangkaian ADC-DAC	66
III.1.7. Pengolah Sinyal Analog	72
III.2. Perencanaan Perangkat Lunak	75
BAB IV. PEMBUATAN DAN UJI COBA	81
IV.1. Pengujian Modul Maksimum Sistem	81
IV.2. Pengujian Modul ADC-DAC	82
BAB V. KESIMPULAN	92
DAFTAR PUSTAKA	93
LAMPIRAN	95



DAFTAR GAMBAR

	Hal
Gambar 2.1. Pin-Pin μ P 8088 Mode Minimum - Mode Maximum	14
Gambar 2.2. Demultiplex Address Bus Microprocessor 8088	17
Gambar 2.3. Bus Timing pada Operasi Read dan Write μ P 8088	21
Gambar 2.4. Read Timing untuk μ P 8088 5 MHz pada Mode Maximum	22
Gambar 2.5. Write Timing untuk μ P 8088 5 MHz pada Mode Maximum	23
Gambar 2.6. Pembangkit Sinyal Clock μ P 8088	24
Gambar 2.7.(a) Pembangkit Sinyal Reset μ P 8088	25
Gambar 2.7.(b) Tegangan Pengisian Kapasitor	26
Gambar 2.8. Sinkronisasi Sinyal RDY dengan Sinyal Clock	27
Gambar 2.9. Register Status Word	30
Gambar 2.10. Register Control Word	31
Gambar 2.11. Pola Bit Tiap Jenis Data pada 8087	35
Gambar 2.12. Hexadesimal Byte untuk Bilangan -127	38
Gambar 2.13. Pararel Comparator ADC	39
Gambar 2.14. Blok Diagram Half Flash ADC	41
Gambar 2.15. Contoh Rangkaian R/2R Ladder DAC	42
Gambar 2.16. Rangkaian Pengganti jika D3 on	42
Gambar 2.17. Rangkaian Pengganti jika D2 on	43

Gambar 2.18.	Rangkaian Pengganti Sederhana jika D2 on	43
Gambar 2.19.	Output Kontrol Proporsional (K_p negatif)	45
Gambar 2.20.	Respon Kontrol Integral terhadap Error yang Konstan	47
Gambar 2.21.	Output Kontrol Derivative	48
Gambar 2.22.	Output Kontrol Proporsional-Integral	49
Gambar 2.23.	Output Kontrol Proporsional-Derivative	50
Gambar 3.1.	Diagram Blok System Secara Umum	55
Gambar 3.2.	Rangkaian Pembangkit Clock μP 8088	58
Gambar 3.3(a).	Rangkaian Pembangkit Sinyal Wait State	59
Gambar 3.3(b).	Sinkronisasi Waktu Satu Sinyal Wait State	59
Gambar 3.4.	Rangkaian Pengontrol Bus	60
Gambar 3.5.	Rangkaian Buffer Data Address	62
Gambar 3.6.	Memory Mapping Maximum Sistem 8088	63
Gambar 3.7.	Rangkaian Dekoder Memory	64
Gambar 3.8.	Rangkaian Masukan dan Keluaran	66
Gambar 3.9.	Rangkaian Penghasil Tegangan Referensi ADC	67
Gambar 3.10.	Rangkaian Penghasil Tegangan Referensi DAC	68
Gambar 3.11.	Rangkaian ADC	69
Gambar 3.12(a).	Rangkaian Penghasil Pulsa Write ADC	70
Gambar 3.12(b).	Grafik Lebar Pulsa (t_w) terhadap Cext	70
Gambar 3.13.	Rangkaian DAC	71
Gambar 3.14.	Rangkaian Driver Motor DC	73
Gambar 3.15.	Filter Lowpass	74

		xi
Gambar 3.17.	Diagram Alur Program Utama	76
Gambar 3.18.	Diagram Alur Proses Simulasi PID	78
Gambar 4.1.	Grafik respon motor untuk nilai - nilai Kp, Kd, Ki, ST berturut-turut 1, 0, 0, 75%	87
Gambar 4.2.	Grafik respon motor untuk nilai - nilai Kp, Kd, Ki, ST berturut-turut 1, 0, 1, 75%	88
Gambar 4.3.	Grafik respon motor untuk nilai - nilai Kp,Kd, Ki, ST berturut-turut 1, 0, 15, 75%	88
Gambar 4.4.	Grafik respon motor untuk nilai - nilai Kd, Ki, ST berturut-turut 1, 0, 0, 50%	89
Gambar 4.5.	Grafik respon motor untuk nilai - nilai Kp, Kd, Ki, ST berturut-turut 1, 0, 1, 50%	89
Gambar 4.6.	Grafik respon motor untuk nilai - nilai Kp,Kd, Ki, ST berturut-turut 1, 0, 15, 50%	90
Gambar 4.7.	Grafik respon motor untuk nilai - nilai Kp, Kd, Ki, ST berturut-turut 1, 0.03, 1, 50%	90
Gambar 4.8.	Grafik respon motor untuk nilai - nilai Kp,Kd, Ki, ST berturut-turut 1, 0.5, 1, 50%	91

DAFTAR TABEL

	Hal
Tabel 2.1. Karakteristik Input μ P 8088	8
Tabel 2.2. Karakteristik Output μ P 8088	8
Tabel 2.3. Status Bus Cycle	13
Tabel 2.4. Sinyal Kontrol Pada Mode Maksimum 8088	15
Tabel 2.5. Status dan Fungsi Queue dari QS1 dan QS0	16
Tabel 2.6. Fan Out yang dianjurkan untuk berbagai Logik	18
Tabel 2.7. Jenis Data Pada 8087	33
Tabel 2.8. Hubungan Keluaran Biner Dengan Keluaran Komparator	40
Tabel 4.1. Hasil Pengukuran ADC	83
Tabel 4.2. Hasil Pengukuran DAC	84
Tabel 4.3. Statistika data ADC	85

BAB I

PENDAHULUAN

I.1. LATAR BELAKANG

Pada bidang studi elektronika, mikroelektronika dipelajari dan dipraktekkan secara langsung melalui mata kuliah mikroelektronika. Mempergunakan prosessor 8088 yang bekerja pada mode minimum sehingga disebut Minimum Sistem 8088. Melalui rangkaian yang telah distandarkan ini, mahasiswa secara langsung membuat perangkat keras sekaligus perangkat lunak pendukungnya. Mencakup pemahaman tentang struktur internal μP 8088, fungsi tiap-tiap pin-nya, rangkaian penunjang seperti rangkaian clock, rangkaian buffer dan latch, rangkaian memori dan mapping memori. Perangkat lunak yang mencakup pemahaman register-register internal 8088, jenis-jenis pengalamatan, serta pemrograman input-output. Bahasa pemrograman yang dipakai adalah Assembler 8088. Alasan pemakaiannya adalah karena kecepatan dan keefisienannya dalam pemakaian memori.

Penerapan Minimum Sistem ini sangat luas pada rangkaian yang memakai mikroprosessor sebagai komponen yang utama. Dengan mikroprosessor 8088 sebagai unit pemroses utama, proses pengontrolan maupun pengukuran dapat dilakukan dengan kecepatan dan ketelitian yang tinggi. Kelebihan lainnya adalah sifatnya yang dapat diprogram ulang sehingga tidak memerlukan penggantian

perangkat keras yang berulang-ulang.

I.2. PERMASALAHAN

Operasi numerik dengan μP 8088 memiliki keterbatasan-keterbatasan. Terutamanya adalah dalam hal kecepatan dan tingkat ketelitian pada operasi dengan bilangan real. Sedangkan pada Assembler 8088 hanya dikenal empat instruksi numerik yaitu mul (perkalian), div (pembagian), add (penjumlahan), sub (pengurangan). Untuk mendapatkan fungsi sinus, misalnya, diperoleh dari rumus dalam bentuk deret. Lalu direalisasikan lewat urutan instruksi-instruksi assembly dengan kombinasi keempat instruksi di atas.

Perhitungan dengan Math coprocessor dapat secara langsung memakai bilangan real, di samping kemudahan berupa instruksi-instruksi yang berupa fungsi yang siap pakai, seperti fsin, ftan, dan sebagainya. Sehingga dimungkinkan proses dengan kecepatan dan ketelitian yang lebih tinggi.

I.3. PEMBATAAN MASALAH

Untuk memanfaatkan Math Co-processor 8087 secara maksimal dibutuhkan pengetahuan pemrograman yang cukup sulit. Dalam tugas akhir ini, prosessor ini akan dipakai untuk mensimulasikan proses PID (Proportional, Integral, Derivatif). Dengan aplikasi pada proses

pengaturan kecepatan motor DC. Hasil perencanaan menentukan nilai penguatan proporsional, integral, dan derivatif yang sangat menentukan ketepatan dan kecepatan untuk mencapai kecepatan tertentu.

I.4. TUJUAN

Tujuan tugas akhir ini adalah merencanakan dan membuat Maksimum Sistem μP 8088 dengan menerapkan Math Co-processor 8087. Diaplikasikan untuk simulasi pengaturan kecepatan motor DC dengan kontrol PID. Sehingga dapat ditentukan nilai penguatan proportional, integral, dan derivative yang tepat untuk motor DC tersebut.

I.5. METODOLOGI

Metodologi tugas akhir ini adalah sebagai berikut:

Mula-mula dilakukan studi literatur mengenai μP 8088 secara perangkat keras. Meliputi struktur internal μP 8088, konfigurasi tiap-tiap pin-nya, karakteristik dc dan ac, mode operasi Minimum dan Maksimum dari μP 8088, operasi read/write memory dan I/O dari μP 8088.

Dipelajari juga μP 8088 dalam suatu sistem. Yang meliputi pembangkitan sinyal timing, interface rangkaian buffer dan latch, interface rangkaian memory, rangkaian pengatur bus, rangkaian pembangkit wait state untuk tiap-tiap satu siklus bus. Bersamaan dengan ini dipelajari juga konfigurasi Math Co-processor 8087

secara perangkat keras. Selanjutnya dipelajari tentang pembebanan pada IC TTL yaitu fan in dan fan out. Dengan selesainya studi ini, maka modul Maksimum Sistem segera dapat direncanakan diatas kertas untuk direalisasikan.

Untuk modul aplikasi, dilakukan studi literatur mengenai karakteristik IC Flash ADC 0820 serta DAC 0830 (pasangan analog to digital dan digital to analog konverter dengan kecepatan tinggi). Juga dipelajari tentang IC Op-Amp sebagai pengolah sinyal analog.

Dalam pembuatan software dipelajari tentang assembler 8088 serta pemrograman Math Co-processor 8087. Sedangkan untuk proses simulasi dipelajari juga mengenai sistem kontrol PID.

I.6. SISTEMATIKA PEMBAHASAN

Sistematika pembahasan tiap bab untuk tugas akhir ini sebagai berikut:

- BAB I - PENDAHULUAN : Bab ini berisi penjelasan mengenai latar belakang permasalahan, permasalahan yang harus dipecahkan, maksud dan tujuan dibuatnya peralatan ini, metodologi dalam pembuatan peralatan, sistematika pembahasan untuk laporan tugas akhir ini, serta relevansi dan manfaat tugas akhir ini.
- BAB II - TEORI PENUNJANG : Bab ini dibagi menjadi tiga bagian inti. Bagian pertama adalah uraian tentang Mikroprosessor 8088 secara perangkat keras. Diawali

dengan uraian blok diagram internal μP 8088, konfigurasi pin-pin μP 8088 beserta fungsi tiap-tiap pin tersebut, mode operasi minimum dan maksimum μP 8088. Selanjutnya dibahas tentang siklus bus, yaitu memori read/write dan I/O read/write. Diikuti dengan uraian rangkaian penunjang μP 8088 dalam suatu sistem. Yang terdiri atas rangkaian pembangkit pewaktu (timing), rangkaian buffer dan latch, rangkaian memori, rangkaian pengontrol bus, rangkaian pembangkit wait state, yang bersama-sama sebagai suatu sistem yang disebut Maksimum Sistem. Terakhir diuraikan tentang konfigurasi Math Co-processor 8087. Bagian kedua adalah penjelasan mengenai konversi dari Analog ke Digital dan sebaliknya. Penjelasan pada bagian ini ditekankan pada kedua IC untuk keperluan itu yang memiliki kecepatan cukup tinggi. Bagian ketiga atau yang terakhir untuk bab ini berisi penjelasan tentang proses kontrol.

- **BAB III - PERENCANAAN** : Pada bab ini akan dijelaskan mengenai perencanaan rangkaian dan komponen-komponen yang dipakai di dalam pembuatan tugas akhir ini. Bab ini dibagi menjadi dua bagian utama, yaitu perencanaan perangkat keras dan perangkat lunak. Pada bagian pertama akan dijelaskan perencanaan maupun pemilihan komponen-komponen yang dipakai pada peralatan tersebut. Sedangkan pada bagian kedua dijelaskan teknik-teknik yang dipakai dalam pembuatan perangkat lunak untuk

peralatan tersebut.

- BAB IV - UJI COBA : Pada bab ini diuraikan secara jelas tentang proses kalibrasi, hasil-hasilnya serta uji coba yang dilakukan dengan pengambilan data.
- BAB V - PENUTUP : Bab terakhir ini berisi kesimpulan dan saran untuk tugas akhir ini.

Bagian-bagian lain yang tidak termasuk dalam kelima bab yang telah dijelaskan di atas tetapi termasuk pendukung dalam laporan tugas akhir ini adalah:

- Kata pengantar, daftar isi, daftar gambar dan daftar tabel yang terletak sebelum kelima bab tersebut.
- Daftar pustaka dan lampiran-lampiran yang terletak pada akhir laporan setelah kelima bab tersebut.

I.7. RELEVANSI

Diharapkan tugas akhir ini dapat dipakai sebagai sarana untuk mempelajari penerapan Math Co-processor 8087 pada aplikasi-aplikasi lain yang dikehendaki.

Sedangkan dari simulasi kontrol PID yang dibuat ini dapat diketahui karakteristik motor DC yang meliputi nilai penguatan proporsional, integral, dan derivatif. Dan diharapkan mempermudah perencanaan dalam pembuatan kontrol PID yang sebenarnya.

BAB II

TEORI PENUNJANG

II.1. Tinjauan Umum μ P 8088

Mikroprocessor 8088 adalah mikroprocessor kedua keluaran INTEL. Secara internal adalah mikroprocessor 16 bit. Tapi secara eksternal μ P 8088 merupakan μ P 8 bit. Memiliki kemampuan pengalamatan sampai 1 Mega byte melalui 20 bit address bus. Sedangkan untuk jalur masuk atau keluar mampu dialamati sampai 64 Kilo byte.

- KARAKTERISTIK DC MIKROPROCESSOR 8088

Mikroprocessor 8088 memerlukan catu daya +5V dengan toleransi 10 persen. Membutuhkan arus maksimum 340 mA, dan bekerja pada temperatur ambient antara 32-180 derajat Fahrenheit. Versi CMOS, yaitu 80C88 dapat beroperasi pada rentang suhu yang lebih lebar dan hanya membutuhkan arus 10mA dari catu daya.

Karakteristik Input. Karakteristik input mikroprocessor ini kompatible dengan semua logik standar komponen-komponen yang ada saat ini. Tabel 2.1 menunjukkan level tegangan input dan arus yang diperlukan tiap-tiap pin. Kebutuhan arus yang rendah ini disebabkan pemakaian gate MOSFET pada pin input.

Karakteristik Output. Tabel 2.2 menunjukkan karakteristik output. Level logika '1' kompatible dengan sebagian besar keluarga IC yang ada sekarang.

Tapi level logika '0' tidak. Logika standar memiliki logika '0' maksimum 0.4V, sedangkan pada mikroprocessor 8088 sebesar 0.45V. Terdapat perbedaan sebesar 0,05V. Perbedaan ini mengurangi 'noise immunity' dari 400 mV pada logika standar menjadi 350 mV. (Noise immunity adalah perbedaan antara logika '0' tegangan output dengan logika '0' tegangan input).

Tabel 2.1. Karakteristik input μP 8088¹⁾

Logik level	Tegangan	Arus
0	0.8V max	10 μ A max
1	2.0V min	10 μ A max

Tabel 2.2. Karakteristik Output μP 8088²⁾

Logik level	Tegangan	Arus
0	0.45V max	2.0mA max
1	2.4V min	-400 μ A max

Mikroprocessor 8088 tersusun atas 40 kaki. Banyak diantaranya yang dimultipleks terhadap waktu. Atau memiliki fungsi ganda. Contohnya adalah jalur address A0

1) Brey, Barry B., The Intel Microprocessors: 8086/8088, 80186, 80286, 80386, and 80486: architecture, programming, and interfacing, Macmillan Publishing Company, 1991, hal 144.

2) ibid, hal 144.

- A7 dengan jalur data D0 - D7, sehingga diberi nama ADO-AD7.

Secara umum konfigurasi kaki-kaki uP 8088 terdiri atas tiga bagian, yaitu : jalur address, jalur data, dan sinyal kontrol.

- KONFIGURASI KAKI-KAKI 8088

- ADO-AD7, Address/Data Bus : jalur yang merupakan multipleks address dan data. Jika ALE aktif ('1') maka jalur ini adalah address memory atau I/O A7-A0, jika ALE tidak aktif ('0') jalur ini adalah data. Selama 'hold acknowledge' jalur ini dalam keadaan high-impedance.

- A15-A8, Address Bus: jalur address A15-A8. Tetap ada selama 'bus cycle'. High impedance saat 'hold acknowledge'.

- A19/S6, A18/S5, A17/S4, A16/S3, Address/Status : jalur multipleks yang menyatakan Address A19-A16 saat ALE aktif. Setelah ALE tidak aktif, sebagai bit status S6-S3 sampai 'bus cycle' berakhir. High-impedance saat 'hold acknowledge'.

- RD, Read : sinyal yang berlogika '0' jika jalur data menerima data dari/ke memory atau I/O. Kaki ini high-impedance saat 'hold acknowledge'.

- READY, ready : jika aktif ('0') akan menyebabkan uP 8088 memperpanjang bus cycle dengan 'wait state'. Ready digunakan pada memory atau I/O dengan access time

yang lebih lambat daripada access time 8088.

- **INTR, Interrupt Request** : pin yang dipakai untuk membangkitkan interupsi secara perangkat keras. Interupsi ini dapat di-enable atau disable melalui IF bit melalui program. Jika pin ini berlogika '1' dan IF diset, 8088 memasuki siklus interupsi (interrupt acknowledge cycle) setelah instruksi yang sedang dieksekusi selesai.

TEST, test : pin yang dicek oleh instruksi Wait. Jika TEST berlogika '0', instruksi akan berjalan normal. Sedangkan jika berlogika '1', maka instruksi WAIT akan menunggu sampai TEST berlogika '0'.

- **NMI, Nonmaskable Interrupt** : input yang akan membangkitkan interupsi type 2, jika aktif. Pin ini aktif saat perubahan dari '0' ke '1', dan tidak dapat di-disable oleh IF bit.

- **RESET, reset** : pin yang bila dijaga berlogika '1' minimal selama empat clock, akan mereset 8088. uP 8088 akan kembali ke nilai-nilai awal dan memulai eksekusi dari alamat FFFF0h.

- **CLK, Clock** : input yang menghasilkan sinyal pewaktuan untuk uP 8088. Memiliki duty cycle 33% (berlogika high selama 1/3 periode clock, dan berlogika low selama 2/3 periode).

- **MN/MX, Minimum/Maximum** : pin untuk menentukan mode operasi uP 8088. Jika dihubungkan langsung ke +5V,

8088 bekerja pada mode minimum. Jika dihubungkan ke ground bekerja pada mode maksimum.

- **BHE/S7, Bus High Enable/Status** : dipakai untuk mengaktifkan jalur data pada orde yang lebih tinggi (D8-D15) selama operasi baca/tulis.

II.1.1. Mode Minimum 8088

Mode minimum 8088 diperoleh dengan menghubungkan pin MN/MX secara langsung ke +5V. Mode minimum ini adalah merupakan cara yang paling sederhana dari operasi uP 8088. Dimana uP 8088 merupakan mikroprocessor tunggal dalam sistem yang dibuat. Mode ini juga paling murah dalam biaya. Semua sinyal kontrol data dan address untuk memory atau I/O dibangkitkan oleh uP 8088 sendiri.

Pin-pin pada mode minimum ini sebagai berikut :

- **IO/M, Input/Output atau Memory** : pin yang menyatakan apakah jalur address mengandung informasi pengalamatan memory atau pengalamatan I/O. Pin ini high-impedance saat hold acknowledge.

- **WR, Write** : sinyal strobe yang menyatakan bahwa jalur data 8088 mengandung data valid untuk memory atau I/O. Pin ini high-impedance saat hold acknowledge.

- **INTA, Interrupt Acknowledge** : merupakan sinyal respon terhadap INTR. Selama ada permintaan interupsi, pin INTA berlogika '0'. Menyatakan bahwa jalur bus

menunggu data angka vektor untuk mendapatkan alamat vektor interupsi.

- **ALE, Address Latch Enable** : pin yang menyatakan bahwa address bus mengandung alamat valid memory atau I/O. ALE tidak pernah berada dalam keadaan high-impedance.

- **DT/R, Data Transmit/Receive** : pin yang dipakai untuk mengontrol arah perpindahan data melalui data buffer eksternal. Pin ini menjadi high-impedance saat hold acknowledge.

- **DEN, Data Bus Enable** : sinyal yang menyatakan bahwa address atau data bus mengandung data yang valid. Pin ini high-impedance saat hold acknowledge.

- **HOLD, Hold** : input yang dipakai untuk membangkitkan permintaan akses langsung ke memory (DMA). Saat HOLD aktif, uP 8088 akan memutuskan hubungan data, address, dan jalur control. Pengontrol DMA eksternal akan mengambil alih akses terhadap memory atau I/O.

- **HLDA, Hold Acknowledge** : suatu indikator bahwa pin HOLD telah berada dalam keadaan high (aktif) dan jalur bus dalam keadaan high-impedance.

- **SSO, Status Line** : pin yang dapat dipakai untuk memperoleh status operasi bus cycle (bersama-sama dengan IO/M dan DTR). Seperti terlihat pada tabel 2.3 berikut.

Tabel 2.3. Status bus cycle³⁾

IO/M	DT/R	SSO	Fungsi
0	0	0	interrupt acknowledge
0	0	1	memory read
0	1	0	memory write
0	1	1	keadaan halt
1	0	0	code akses
1	0	1	I/O read
1	1	0	I/O write
1	1	1	keadaan pasif

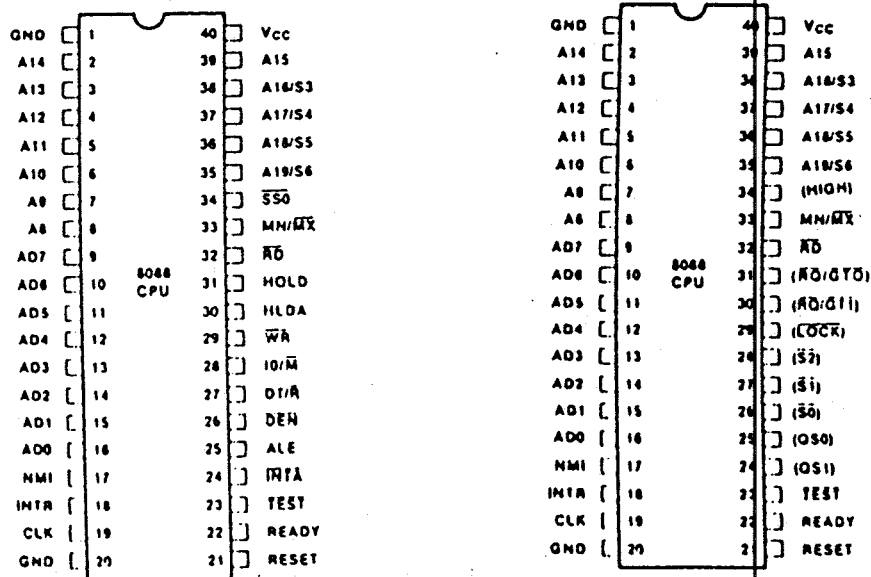
II.1.2. Mode Maksimum μ P 8088

Operasi pada mode maksimum lebih kompleks daripada mode minimum. Mode maksimum dipakai untuk mengaplikasikan lebih dari satu mikroprocessor dalam suatu sistem yang dibuat. Mikroprocessor 8088 memerlukan chip khusus yang terpisah untuk menghasilkan sinyal-sinyal kontrol. Karena beberapa pin-nya telah digantikan oleh pin yang berbeda dengan fungsi dan karakter yang berbeda. Perubahan tersebut akan

³⁾ ibid, hal. 147.

diperlihatkan pada gambar 2.1.

Operasi μP 8088 pada mode maksimum diperoleh dengan menghubungkan pin MN/MX ke ground. Beberapa pin-pin yang terdapat pada mode maksimum ini adalah:



Gambar 2.1. Pin-pin μP 8088 Mode Minimum-Maximum⁴⁾

- S2, S1, S0, Status : bit-bit yang dipakai dalam mode maksimum untuk membangkitkan sinyal kontrol utama melalui chip 8288. Jadi 8288 ini akan mengkodekan pin S0, S1, dan S2 untuk menghasilkan sinyal kontrol yang bersesuaian. Tabel 2.4 berikut menunjukkan kombinasi S0, S1, dan S2 serta sinyal kontrol yang dihasilkannya.

⁴⁾ ibid, hal. 143.

Tabel 2.4. Sinyal Kontrol pada Mode Maksimum 8088⁵⁾

S ₂	S ₁	S ₀	Fungsi kontrol
0	0	0	interrupt acknowledge
0	0	1	I/O read
0	1	0	I/O write
0	1	1	keadaan halt
1	0	0	kode akses
1	0	1	memory read
1	1	0	memory write
1	1	1	keadaan pasif

- RQ/GT0 dan RQ/GT1, Request/Grant : pin yang dipakai untuk menyatakan bahwa eksternal coprocessor hendak mengambil alih bus. Masing-masing pin ini adalah pin dua arah. Sehingga coprocessor bisa melakukan permintaan DMA, 8088 dapat mengijinkan terjadinya suatu permintaan DMA, dan coprocessor dapat memberitahukan kembali bahwa DMA telah selesai sehingga bus dapat diambil alih lagi oleh 8088.

- LOCK, Lock : sinyal output yang berlogika '0' jika suatu instruksi diawali dengan perintah Lock.

⁵⁾ ibid, hal. 47.

Umumnya dipakai untuk mencegah processor lain mengakses 8088 bus selama instruksi-instruksi tersebut.

-QS1 dan QS0, Queue Status : kedua bit ini dipakai untuk mengetahui kondisi internal queue yang akan terjadi. Tabel 2.5 berikut menunjukkan kombinasi kedua pin ini dan fungsinya.

Tabel 2.5. Status dan Fungsi Queue dari QS1 dan QS0⁶⁾

QS1	QS0	Fungsi queue
0	0	tidak ada operasi
0	1	byte pertama dari queue
1	0	queue dalam keadaan kosong
1	1	byte berikutnya dari queue

- BUS BUFFER DAN LATCH

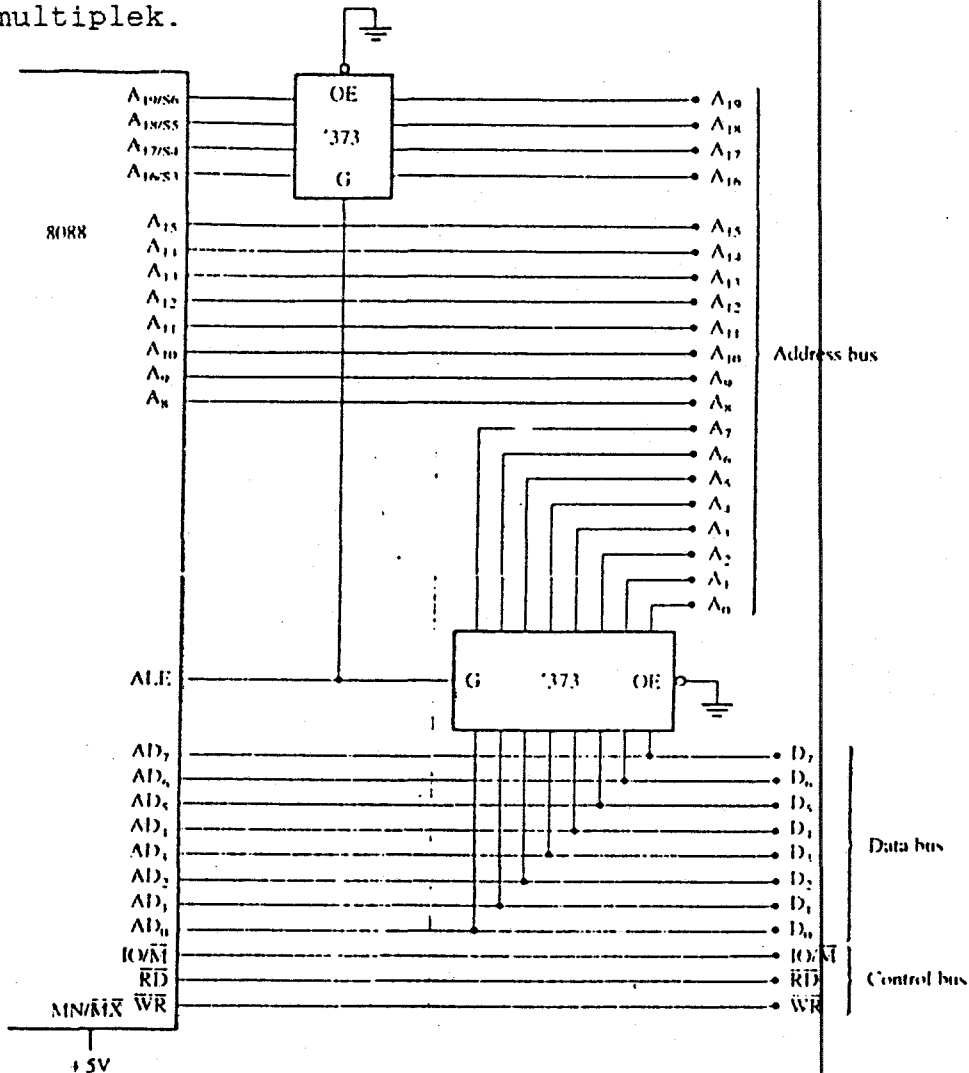
Latch Bus. Sebelum Mikroprocessor 8088 dapat di-interface-kan dengan memori atau I/O, diperlukan suatu rangkaian demultiplek. Rangkaian demultiplek ini diperlukan karena address memori atau I/O harus tetap valid dan stabil selama operasi read/write.

Secara garis besar ada tiga bus yang harus ada dalam suatu rangkaian interface, yaitu: address bus

⁶⁾ ibid, hal. 148.



yang menyediakan alamat memori atau port I/O, data bus yang mentransfer data ke dan dari memori dan I/O, kontrol bus yang menyediakan informasi kontrol ke memori dan I/O. Rangkaian demultiplek ini juga harus berfungsi sebagai latch yang menjaga outputnya tetap stabil. Rangkaian ini diaktifkan oleh sinyal ALE dari μP 8088. Gambar 2.2 memperlihatkan address bus yang di-demultiplek.



Gambar 2.2. Demultiplek addresss bus μP 8088⁷⁾

⁷⁾ ibid, hal. 153.

Buffer Bus. Rangkaian buffer bus diperlukan untuk memperbaiki kemampuan mendrive beban. Sehingga lebih banyak unit beban TTL yang dapat dihubungkan pada mikroprosessor 8088. Tabel 2.6 di bawah memperlihatkan jumlah unit beban yang dianjurkan jika tidak dipakai buffer.

Untuk lebih dari 10 unit beban, maka seluruh pin address/data harus dibuffer.

Tabel 2.6. Fan out yang dianjurkan untuk berbagai logik⁸⁾

Keluarga	Fan out	Arus sink	Arus source
TTL(74)	1	-1.6 mA	40 μ A
TTL(74LS)	5	-0.4 mA	20 μ A
TTL(74S)	1	-2.0 mA	50 μ A
TTL(74ALS)	10	-0.2 mA	20 μ A
CMOS(74HC)	10	-1 μ A	1 μ A
CMOS(CD)	10	-1 μ A	1 μ A
NMOS	10	-10 μ A	10 μ A

Disebut dengan konfigurasi Full Buffer. Sinyal dengan full buffer akan mengakibatkan timbulnya delay.

⁸⁾ ibid, hal. 145.

Sehingga diperlukan perhitungan yang teliti apabila dipergunakan device yang lambat, untuk menentukan perlu atau tidaknya rangkaian wait state.

Bus Timing. Dua macam operasi penting yang akan dibahas adalah operasi read dan write. Jika mikroprosesor melakukan operasi read, mikroprosesor mengeluarkan address pada address bus, membangkitkan sinyal read, dan menerima data melalui data bus. Sedangkan operasi write diawali dengan mengeluarkan address, lalu data, terakhir membangkitkan sinyal write.

Mikroprosesor 8088 memakai memori dan I/O dalam suatu periode waktu yang disebut bus cycle. Satu bus cycle berlangsung selama empat periode clock, T1-T4. Pada clock dengan frekuensi 5 MHz, satu bus cycle sama dengan 800 ns. Proses yang terjadi pada masing-masing periode clock tersebut sebagai berikut :

T1. Selama periode clock pertama, T1, proses yang terjadi adalah : memory atau I/O address dikeluarkan melalui address bus dan jalur multiplek address/data, juga sinyal-sinyal kontrol ALE, DT/R, DAN IO/M.

T2. Selama T2 dibangkitkan sinyal RD atau WR, DEN, dan jika proses write, data yang hendak ditulis juga dikeluarkan. Proses diatas menyebabkan memori atau I/O 'bersiap' memulai operasi read/write. Mikroprosesor mengaktifkan sinyal DEN, sehingga memori atau I/O dapat menerima data atau mikroprosesor dapat menerima data

yang dibaca dari memori atau I/O. Sedangkan jika operasi write, maka data dapat dikeluarkan ke memori atau I/O melalui data bus. Pada akhir T2, sinyal READY dicek, jika berlogika low, disisipkan Tw. Yang berarti terjadi penambahan satu periode untuk wait state.

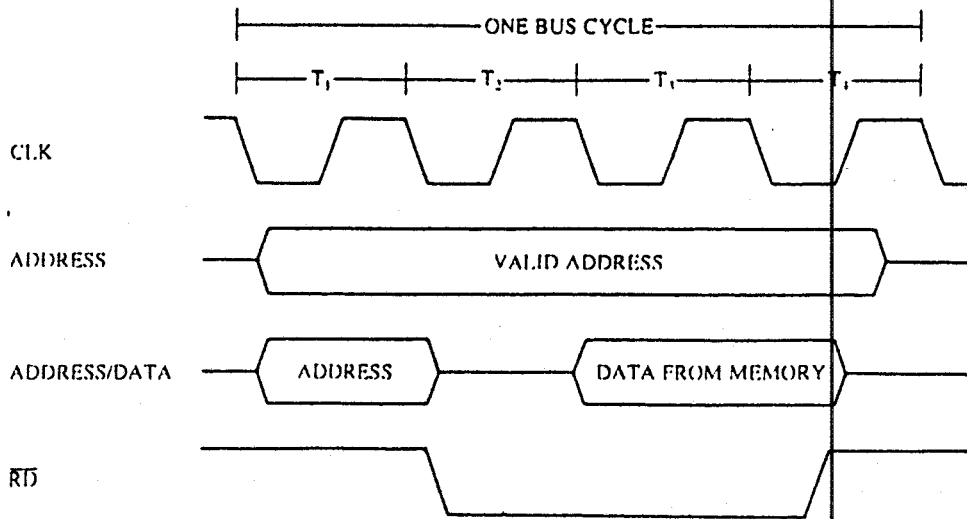
T3. Periode T3 ini disediakan bagi memori atau I/O untuk mengakses data. Jika merupakan operasi read, data disampel pada akhir T3.

T4. Selama T4 semua sinyal dipersiapkan untuk bus cycle berikutnya. Jika operasi read, mikroprosesor mengambil data dari data bus, sedangkan jika operasi write, sisi naik dari sinyal WR akan mentransfer data ke memori atau I/O. Gambar 2.3 akan memperlihatkan bus timing pada operasi read dan write.

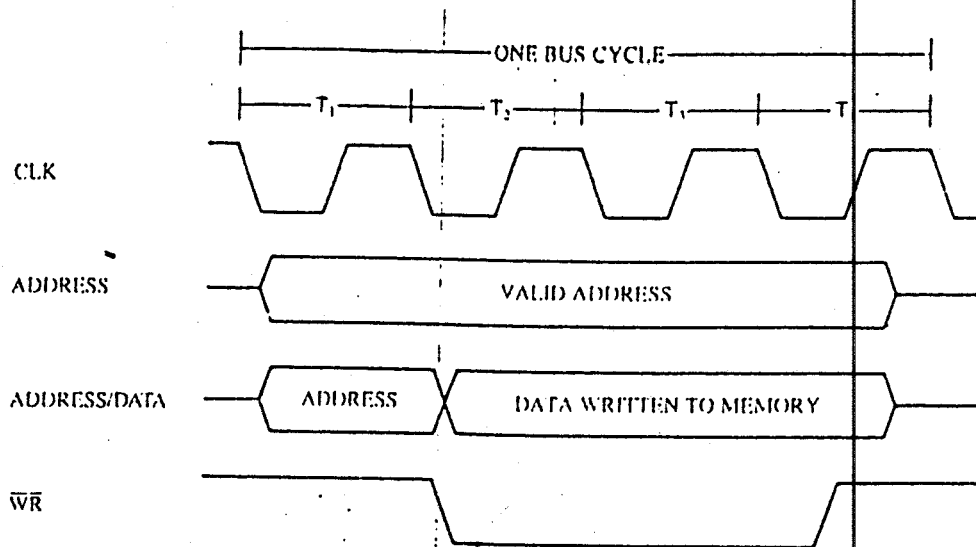
Di bawah ini akan diuraikan point-point penting saja dari timing diagram tersebut yang merupakan karakteristik AC mikroprocessor 8088.

MEMORY READ TIMING. Dua yang terpenting pada operasi read adalah address access time (TAVDV) dan read access time (TRLDV). Address access time adalah waktu dari saat keluarnya memori address pada pin AD0-AD15 dan A16-A19 sampai data valid disampel oleh mikroprocessor. Read access time adalah waktu dari aktifnya sinyal RD (RD low) sampai data valid disampel oleh mikroprocessor. Ini ditentukan oleh waktu akses yang dimiliki memori atau I/O tersebut. Karena adanya delay

pada buffer dan dekoder maka waktu akses berkurang sekitar 30-40 ns.



Gambar 2.3.(a). Bus timing pada operasi read μP 8088⁹⁾

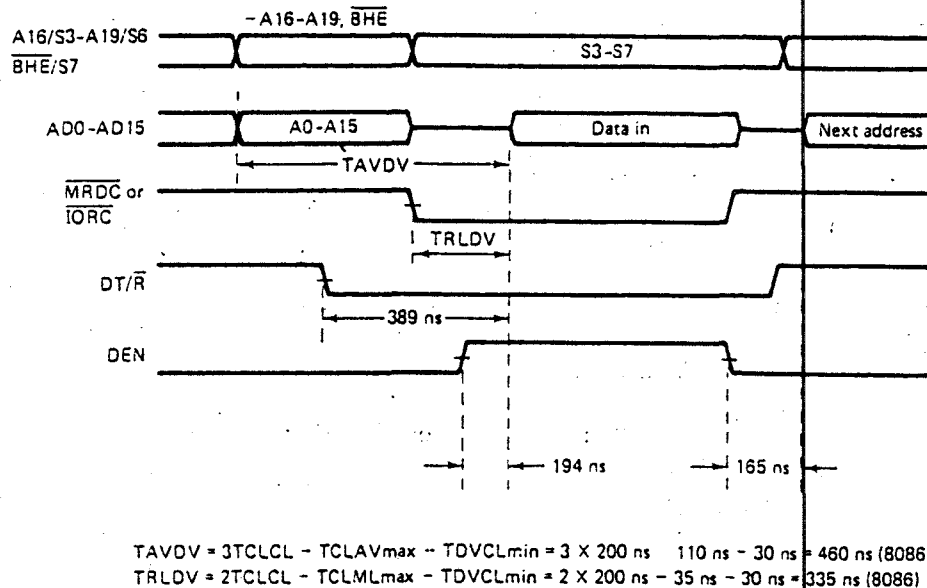


Gambar 2.3.(b). Bus timing pada operasi write μP 8088¹⁰⁾

⁹⁾ ibid, hal. 158.

¹⁰⁾ ibid, hal. 157.

Gambar 2.4. memperlihatkan address dan read access time pada mode maksimum.



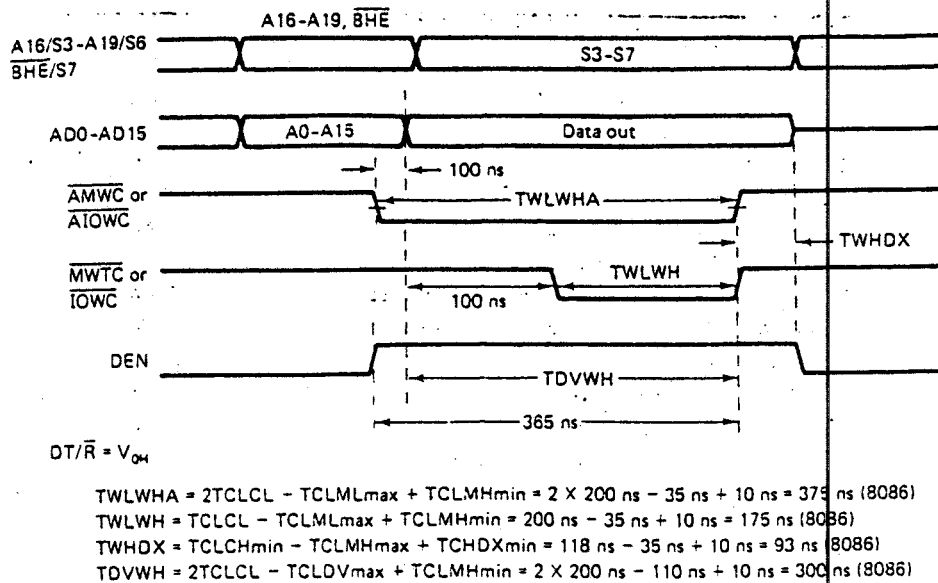
Gambar 2.4. Read timing untuk μP 8088 5 MHz pada mode maksimum.¹¹⁾

MEMORY WRITE TIMING. Dua hal yang terpenting pada operasi write adalah memori setup time (TDVWH) dan data hold time (TWHDX). Memori setup time adalah waktu dari keluarnya data valid sampai sinyal WR berlogika high. Data hold time adalah selang waktu data dijaga konstan pada data bus setelah sinyal WR tidak aktif.

Yang juga perlu diperhatikan adalah lebar pulsa write minimum komponen memori yang dipakai. Gambar 2.5 memperlihatkan memori setup time dan data hold time

¹¹⁾ Uffenbeck, John, The 8086/8088 Family: Design, Programming, and Interfacing, hal. 288.

pada mode maksimum.



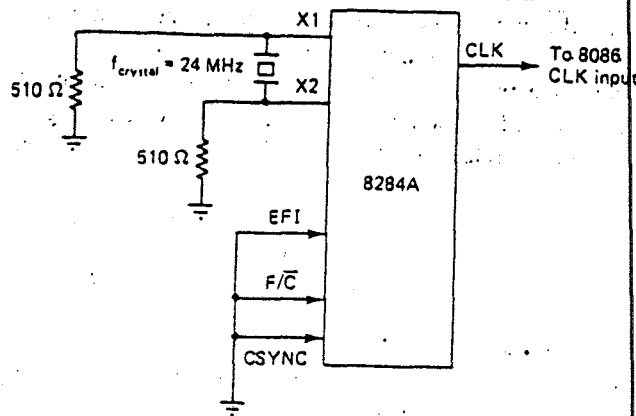
Gambar 2.5. Write timing untuk μP 8088 5 MHz pada mode maksimum¹²⁾

- SISTEM CLOCK

Mikroprocessor 8088 membutuhkan sinyal clock dengan rise time dan fall time yang cepat ($< 10 \text{ ns}$). Level logika '0' dan '1' masing-masing sebesar $-0.5 - 0.6 \text{ V}$ dan $3.9 - 5.0 \text{ V}$ dengan duty cycle 33%. Karena kebutuhan yang spesifik ini, maka INTEL juga mengeluarkan chip khusus untuk memenuhi kebutuhan-kebutuhan seperti yang disebutkan di atas, yaitu 8284. Chip ini juga dilengkapi dengan sinyal reset serta pembangkit sinyal wait state bagi device yang lambat. Gambar 2.6 di bawah memperlihatkan pembangkitan sinyal

¹²⁾ ibid, hal. 289.

clock dengan menggunakan chip 8284.

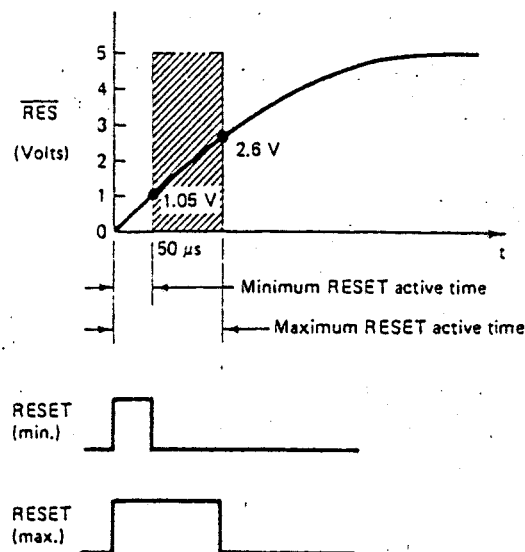
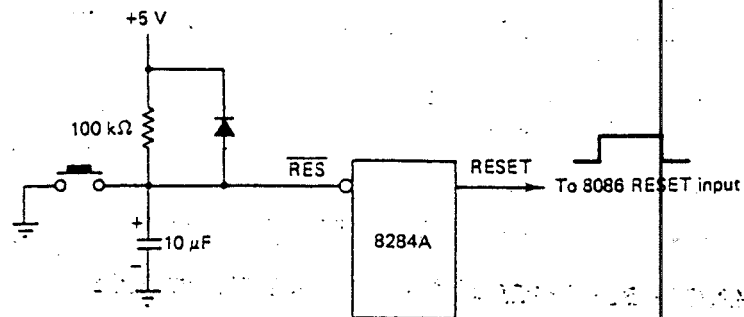


Gambar 2.6. Pembangkit sinyal clock μP 8088¹³⁾

Sinyal reset memerlukan lebar minimum sebesar 50 μs . Ini bisa diperoleh dengan memberi rangkaian RC seri pada pin RES 8284. Gambar 2.7 di bawah ini memperlihatkan rangkaian RC yang ditambahkan pada input RES 8284 beserta sinyal reset yang dihasilkannya. Dengan cara seperti itu, mikroprocessor dapat direset dengan dua cara. Yaitu secara otomatis pada saat power dipasang dan secara manual dengan menekan switch selama beberapa saat lalu melepaskannya. Ketika power supply dipasang, kapasitor akan mengisi. Dimana lama pengisian ini ditentukan oleh time konstan $\sigma = R.C$. Jadi dengan memilih nilai R dan C yang sesuai akan diperoleh sinyal reset yang memenuhi persyaratan di atas. Diode pada rangkaian di bawah berfungsi sebagai jalur pembuangan

¹³⁾ ibid, hal. 233.

muatan kapasitor begitu sistem dimatikan (power dilepas).

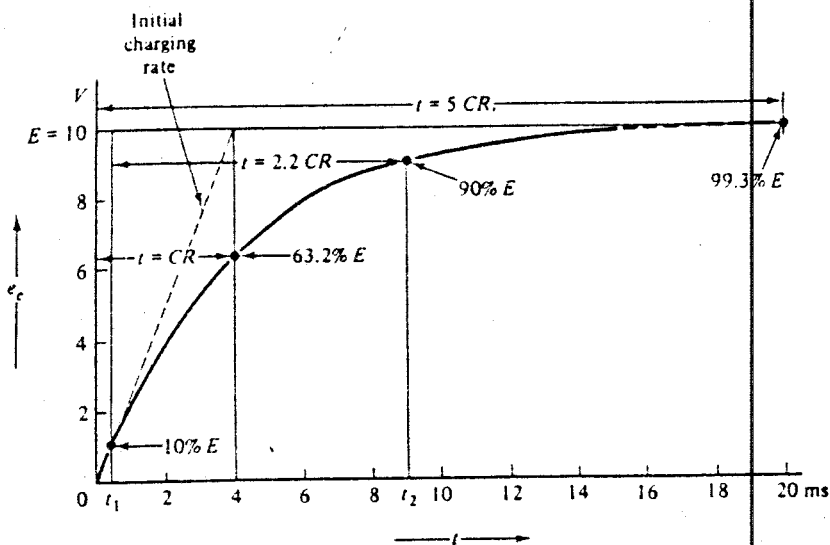


Gambar 2.7.(a). Pembangkit sinyal reset μP 8088¹⁴⁾

Pembangkitan sinyal reset secara manual terjadi ketika saklar ditekan. Kapasitor terhubung singkat dan dalam waktu $t \approx 0$ muatannya telah habis terbang ke ground. Sehingga pada saat saklar dilepas, kapasitor

¹⁴⁾ ibid, hal. 235.

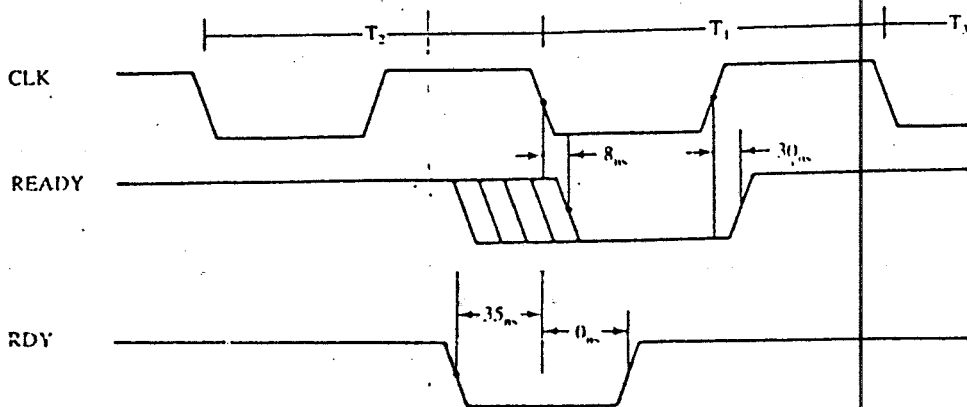
akan mengisi dengan time konstan seperti pada reset secara otomatis. Jadi pada dasarnya reset secara otomatis sama saja dengan reset secara manual. Karena proses yang terjadi sama, yaitu proses pengisian kapasitor dari keadaan tanpa muatan menuju ke V_{cc} .



Gambar 2.7.(b). Tegangan pengisian kapasitor¹⁵⁾

Pembangkitan sinyal wait state ditentukan oleh pin-pin input RDY dan AEN. Sinyal pada pin-pin ini harus sinkron dengan clock untuk menghasilkan sinyal wait yang disampel pada akhir T2 dari siklus bus yang berlangsung. Gambar 2.8 memperlihatkan sinkronisasi sinyal RDY serta sinyal Ready dengan clock.

¹⁵⁾ Bell, A.David, Solid State Pulse Circuit, Reston Publishing Company Inc., 2nd edition, hal. 37.



Gambar 2.8. Sinkronisasi sinyal RDY-Ready dengan clock¹⁶⁾

II.2. Math Co-Processor 8087

Untuk peralatan-peralatan tertentu dibutuhkan pengolahan data dengan perhitungan yang cepat. Dengan adanya 8087 maka perhitungan yang biasanya memerlukan beberapa instruksi pada μP 8088 dapat dikerjakan hanya oleh satu instruksi pada 8087. Sehingga didapat kecepatan yang lebih tinggi, khususnya untuk perhitungan-perhitungan yang cukup rumit. Kelebihan yang lainnya adalah kemampuan untuk bekerja dengan bilangan real.

- ORGANISASI CO-PROCESSOR

μP 8087 dirancang untuk dipakai bersama dengan μP

¹⁶⁾ Brey, Barry B., *op.cit.*, hal. 163.

8088. Melakukan pembacaan instruksi yang terdapat dalam Quene. Tetapi 8087 hanya melaksanakan instruksi yang memang ditujukan padanya, dan mengabaikan instruksi untuk 8088. Sebaliknya 8088 juga membaca instruksi untuk 8087 yang berupa instruksi Escape, tapi mengabaikannya tetapi tetap melakukan pembacaan terhadap operand-operandnya. Operand ini kemudian dipakai oleh 8087 untuk melakukan pembacaan atau penulisan data dari atau ke memory.

Untuk menjamin terjadinya sinkronisasi dari 8087 dan 8088 yang bekerja paralel, program-program harus mengikuti aturan sebagai berikut :

- μ P 8088 tidak boleh melakukan perubahan terhadap isi memory yang dipakai dalam instruksi 8087 sampai 8087 selesai mengerjakan instruksi tersebut. μ P 8088 dapat mengubah register dan flag-nya sendiri.
- Instruksi 8087 yang berikutnya tidak boleh dibaca oleh 8088 sampai instruksi yang sedang dikerjakan oleh 8087 selesai dilakukan.

Dalam hal ini, pembuat program bertanggung jawab untuk memperhatikan bahwa aturan pertama dipenuhi. Karena kerap kali instruksi 8087 diikuti oleh instruksi berikutnya, yang merupakan instruksi 8088. Sehingga untuk menjamin terjadinya sinkronisasi, maka programmer wajib memasukkan instruksi FWAIT sesudah instruksi 8087 dan sebelum instruksi 8088. Apabila kedua instruksi

tersebut mengakses lokasi memory yang sama. Sedangkan untuk memenuhi aturan kedua, tanggung jawab berada pada Assembler yang dipakai. Adalah kewajiban Assembler untuk secara otomatis meletakkan instruksi WAIT di depan setiap instruksi 8087. Dengan demikian, 8088 tidak akan mengambil instruksi 8087 yang berikutnya.

- REGISTER INTERNAL 8087

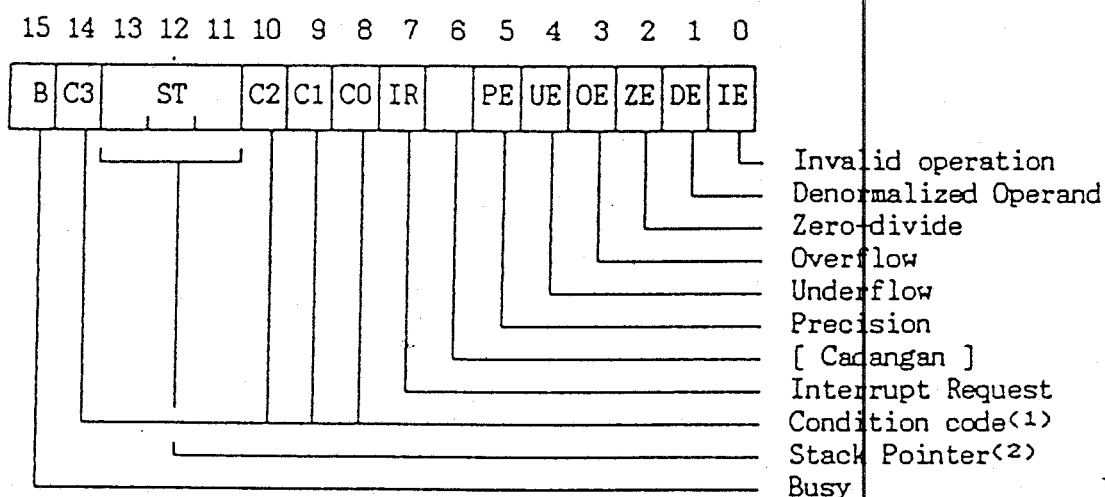
Lima register internal yang dimiliki oleh 8087 dapat diakses oleh programer. Kelima register tersebut adalah: register stack, status word, control word, tag word dan pointer exception.

Program perhitungan MPU 8087 terdiri dari delapan register 80 bit. Kedelapan register ini membentuk stack, dan biasa disebut register stack. Register atau stack yang paling atas disebut ST atau ST(0). Register berikutnya adalah ST(1), dan seterusnya sampai ST(7). Beberapa instruksi 8087 yang tidak mengandung operand menggunakan ST(0) sebagai register untuk operasinya. Sehingga dapat dikatakan ST(0) memiliki kedudukan seperti register AX (Accumulator) pada 8088. Struktur organisasi kedelapan stack tersebut membentuk suatu lingkaran, sehingga dibawah ST(7) adalah ST(0). Oleh karena itu programer bertanggung jawab untuk menghindari terjadinya Stack Overflow.

Status word yang terdiri dari 16 bit data menunjukkan keadaan/status yang dipengaruhi oleh

operasi 8087. Gambar 2.9 memperlihatkan fungsi tiap bit dari status word.

Control word 16 bit pada 8087 memungkinkan berbagai fungsi yang dapat dipilih dan ditentukan oleh program. Fungsi-fungsi tersebut antara lain exception enable mask dan interrupt enable mask yang cukup penting bagi programmer. Selain itu fungsi yang lain adalah penentuan letak pembulatan, kontrol untuk infinity dan ketelitian yang berguna untuk operasi-operasi numerik.



Keterangan :

(1) Dijelaskan pada bagian selanjutnya

(2) Nilai ST :

000 menunjuk ke register 0

001 menunjuk ke register 1

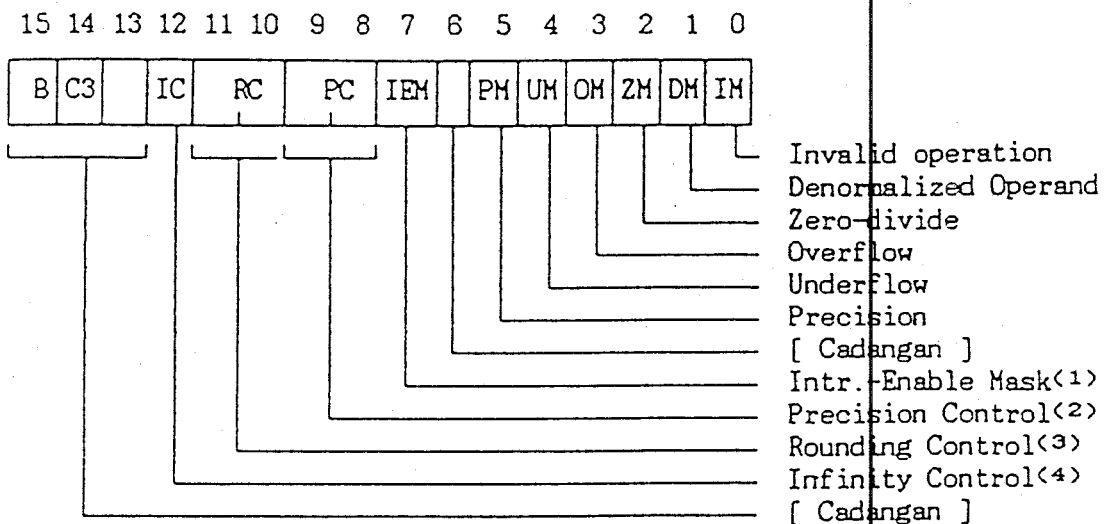
•

111 menunjuk ke register 7

Gambar 2.9. Register Status Word¹⁷⁾

¹⁷⁾ Startz, Richard, 8087 Application and Programming for the IBM PC and Others PCs, Prentice-Hall Publishing, California, 1983, hal. 27.

Tag word mempunyai dua bit untuk tiap register pada stack yang menandakan apakah register tersebut valid, nol, spesial atau kosong. Sedangkan Pointer Exception berisi instruksi dan operand yang sedang dikerjakan oleh 8087. Kedua jenis register yang disebutkan terakhir ini tidak penting bagi pemrogram.



Keterangan :

- (1) Interrupt-Enable Mask :
 - 0 = Interrupt Enable
 - 1 = Interrupt Disable
- (2) Precision Control :
 - 00 = 24 bit
 - 01 = (cadangan)
 - 10 = 53 bit
 - 11 = 64 bit
- (3) Rounding Control :
 - 00 = Pembulatan ke tempat terdekat
 - 01 = Round Down (Pembulatan ke bawah/arrah negatif)
 - 10 = Round Up (Pembulatan ke atas/arrah positif)
 - 11 = Chop (Pembulatan ke arah angka nol)
- (4) Infinity control :
 - 0 = Projective
 - 1 = Affine

Gambar 2.10. Register Control word¹⁸⁾

¹⁸⁾ ibid, hal. 28.

- CONTROL WORD

Salah satu fungsi yang disediakan oleh Register Control Word yang cukup penting adalah pembulatan. Ada empat macam pembulatan yang disediakan : pembulatan ke tempat terdekat, pembulatan ke atas, pembulatan ke bawah, pembulatan ke arah nol (untuk bilangan positif dibulatkan ke bawah, bilangan negatif dibulatkan ke atas). Sedangkan nilai defaultnya adalah pembulatan ke tempat terdekat.

Tidak seperti kebanyakan komputer, 8087 sanggup melakukan perhitungan yang mengandung nilai tak berhingga. μP 8087 mampu memberikan jawaban yang tepat, bila memang secara matematis nilai dari perhitungan tersebut ada dan terdefinisi. Sebagai contoh, $5/\infty = 0$.

Kontrol untuk bilangan tak berhingga ini memiliki dua mode, Projective dan Affine. Yang dimaksud dengan mode Projective untuk bilangan tak berhingga adalah anggapan bahwa bilangan + dan - adalah sama. Jadi dalam mode ini garis bilangan dianggap berbentuk melingkar sehingga kedua ujungnya bertemu yang disebut 'tak berhingga'. Sedangkan dalam mode Affine, dianggap garis bilangan tersebut mempunyai dua ujung (tak bertemu). Ujung yang satu adalah $-\infty$ dan ujung yang lain $+\infty$. Nilai default untuk ini adalah mode Projective.

Ketelitian pada 8087 dapat dipilih antara : 64, 53, atau 24 bit sesuai dengan jenis data temporary

real, double precision, atau single precision. Yang menjadi defaultnya adalah 64 bit (temporary real).

- SISTEM BILANGAN

Dalam 8087 dikenal 3 sistem bilangan, yaitu : floating point, integer, dan packed decimal. Dari ketiga sistem tersebut yang paling penting adalah floating point.

Floating point merupakan sistem bilangan bagi komputer yang serupa dengan bentuk eksponen. Sebagai contoh nilai "negatif setengah" dapat ditulis dalam bentuk eksponen : -5.0×10^{-1} . Tiga bagian penting dari bentuk tersebut adalah sign (tanda negatif/positif), angka penting, dan eksponen.

Tabel 2.7. Jenis Data Pada 8087¹⁹⁾

Jenis Data	Jumlah bit	Angka penting	Range
Word Integer	16	4	-32768 s/d 32767
Short Integer	32	9	-2×10^9 s/d 2×10^9
Long Integer	64	18	-9×10^{18} s/d 9×10^{18}
Packed Decimal	80	18	18 angka dec + sign
Short Real	32	6 / 7	10^{-37} s/d 10^{38}
Long Real	64	15 / 16	10^{-307} s/d 10^{308}
Temporary Real	80	19	10^{-4932} s/d 10^{4932}

¹⁹⁾ ibid, hal. 30.

Tabel 2.7 memberikan jumlah bit yang dipakai untuk menyimpan suatu jenis data, jumlah angka penting, serta range untuk setiap jenis data.

- 8087 DATA BIT

MPU 8087 mengenal suatu jenis data secara tepat dengan memperhatikan fungsi tiap-tiap bit secara tepat. Urut-urutan fungsi tiap bit ini disesuaikan dengan hardware dari 8087, dan dapat membingungkan pemrogram. Tapi jika suatu ketika dilakukan debugging terhadap program dalam bahasa Assembly, maka perlulah diketahui secara tepat fungsi tiap-tiap bit data tersebut.

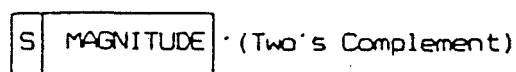
Pada gambar 2.11 akan diperlihatkan fungsi tiap bit data 8087. Perlu diperhatikan bahwa cara menyimpan bit tersebut pada memory adalah sama seperti cara komputer menyimpan, double word dan seterusnya. Sebagai contoh jika komputer menyimpan bilangan integer yang memiliki jumlah bit 16 pada lokasi memory 100 dan 101. Pada lokasi memory 100 akan disimpan bit 0 - 7, kemudian pada lokasi memory 101 disimpan bit 8 - 15.

Untuk Floating Point, 8087 memiliki cara khusus dalam menyimpan nilai-nilai bit tersebut. Berikut ini adalah aturan penyimpanan nilai bit untuk floating point.

- Seperti kebanyakan angka pada komputer, maka untuk jenis Floating Point ini tiap besaran di dalamnya dinyatakan dalam binary biasa, dan bukan packed

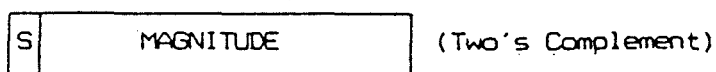
desimal.

WORD INTEGER



15 0

SHORT INTEGER



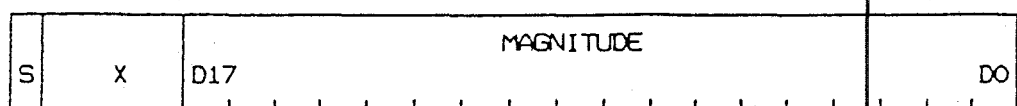
31 0

LONG INTEGER



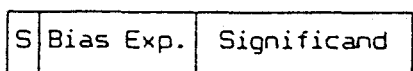
63 0

PACKED DECIMAL



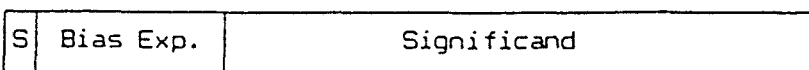
79 72 0

SHORT REAL



31 23 0

LONG REAL



63 52 0

TEMPORARY REAL



79 64 0

Keterangan :

S = bit tanda (sign), 0 = positif, 1 = negatif

X = don't care (diabaikan)

Dn = angka desimal (1 nibble)

Gambar 2.11. Pola bit tiap jenis data pada 8087²⁰⁾

²⁰⁾ ibid, hal. 33.

- Posisi point binary tidak dinyatakan. Sebagai gambarannya misalkan desimal 153.7 yang dapat dinyatakan sebagai 1.537E2. Jadi dapat dikatakan untuk bentuk eksponen, point decimal selalu terletak disebelah kanan MSB-nya. Begitu pula pada sistem binary, dapat dianggap bahwa letak point binary-nya selalu di sebelah kanan MSB sehingga tidak perlu dinyatakan keberadaannya.
- Angka Floating Point selalu dinyatakan dalam bentuk yang "ternormalisir". Bit permulaan dari angka tersebut selalu "1". Bila perlu komputer akan menggeser angka penting (significant), sementara mengurangi atau menambah nilai eksponen untuk mendapatkan bentuk ternormalisir tersebut.
- Karena selalu diawali "1", maka komputer tidak menyimpannya untuk meningkatkan ketelitian.
- Eksponen dapat bernilai positif atau negatif. Untuk menyatakan ini komputer tidak menggunakan sign bit, melainkan langsung dengan menjumlahkan nilai eksponen sesungguhnya dengan suatu konstanta. Contohnya untuk angka real single-precision nilai eksponen yang sesungguhnya dijumlahkan dengan 127 sebelum disimpan oleh komputer. Untuk double precision konstantanya adalah 1023 dan untuk temporary real 16383.
- Angka nol dinyatakan dengan semua bit eksponen serta bit significant dibuat "0". Bit tanda diabaikan.

Contoh untuk menyatakan floating point : angka 2.0 memiliki significant bit "[1]00...". Tanda kurung menyatakan bahwa angka "1" tidak disimpan. Di situ juga terlihat bahwa komputer tidak memperhatikan letak point binary. Eksponen bitnya adalah $128(127+1)$.

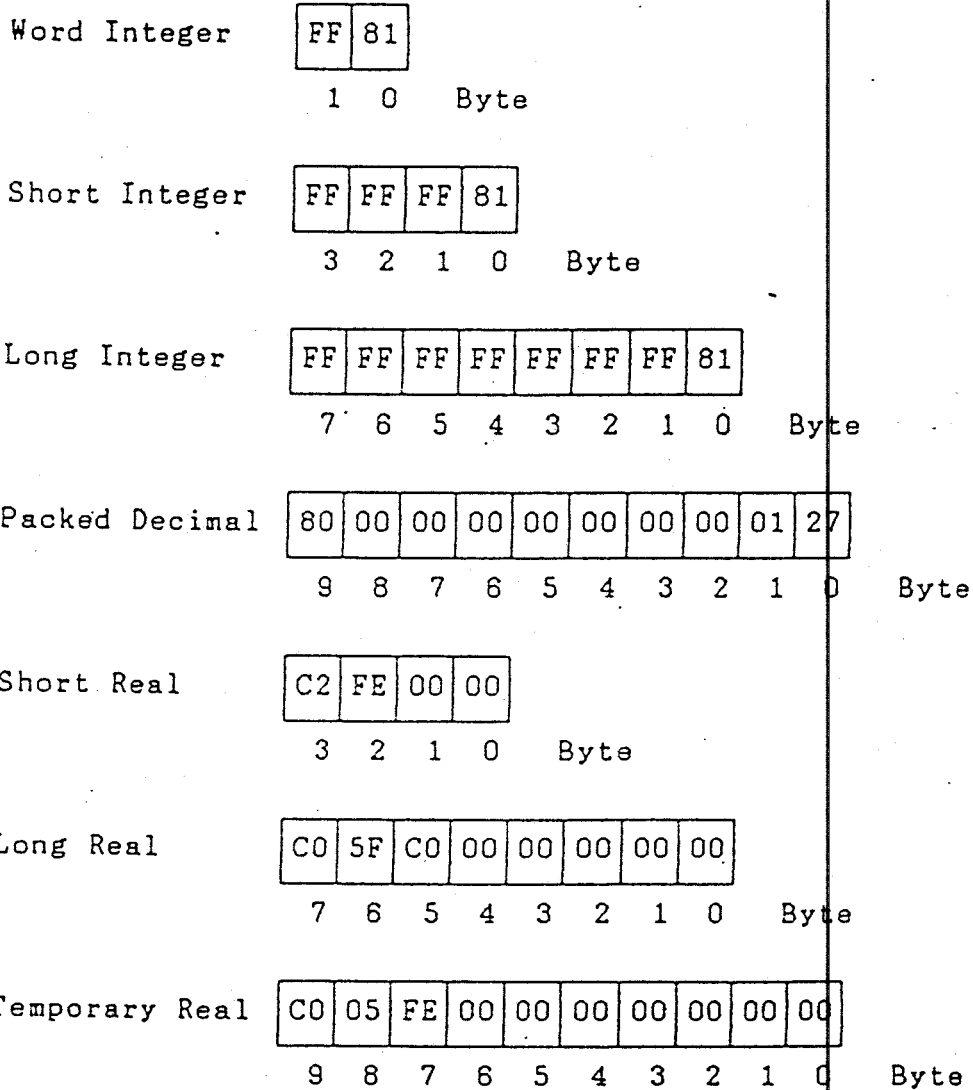
Sedangkan untuk angka $\frac{1}{2}$ yang dapat dinyatakan sebagai 2^{-1} , significant bitnya "[1]00...". Sama dengan significant 2.0 pada contoh sebelumnya. Eksponen bitnya adalah 126 ($127-1$). Sedangkan untuk angka 3 dan 4 significant bitnya masing-masing "[1]10..." dan "[1]00..." dan eksponen bitnya masing-masing adalah 128 dan 129.

Untuk bentuk integer, angkanya dinyatakan dalam "two's complement". Untuk bilangan positif maka bentuknya adalah binary biasa. Sedangkan untuk bilangan negatif maka dinyatakan dengan cara : Jika X adalah bagian positifnya, maka $-X$ ditulis dalam bentuk $(NOT X)+1$. Dengan bentuk "two's complement" ini, maka bit paling kiri dari bilangan negatif selalu "1" dan pada bilangan positif selalu "0".

Sedangkan bentuk Packed Decimal, langsung dinyatakan dalam tanda dan memiliki jumlah angka 18. Bit 0-3 menampung angka desimal yang pertama, bit 4-7 angka yang kedua dan seterusnya. Bit 72-78 tidak dipakai.

Pada gambar 2.12 diberikan contoh penerapan untuk

tiap jenis data yang menyatakan nilai -127 dalam bentuk hexadecimal byte. Format untuk tiap jenis data tersebut sesuai dengan format yang telah dijelaskan sebelumnya.



Gambar 2.12. Hexadesimal byte untuk bilangan -127²¹⁾

²¹⁾ ibid, hal. 35.

II.3. Konversi Analog Digital

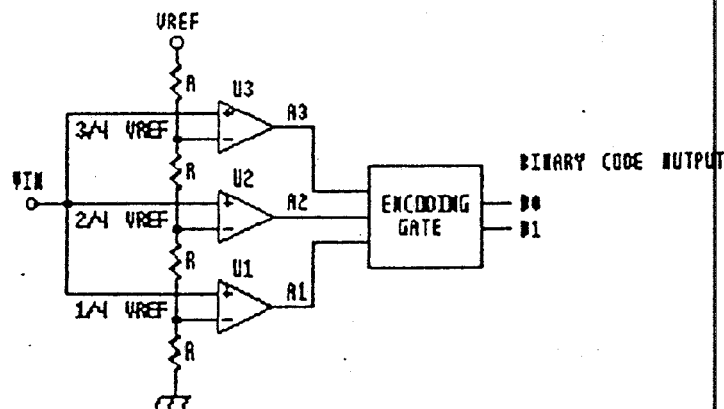
Pengubahan data analog menjadi digital atau data digital menjadi analog merupakan kunci utama dari sistem digital dalam melakukan proses. Pengubahan data ini dilakukan dengan cara dan sistem yang beragam. Masing-masing memiliki keunggulan dan kelebihan.

II.3.1. Konversi Analog ke Digital

Banyak IC dewasa ini yang dapat langsung mengubah besaran analog ke digital. Karena dalam tugas akhir ini dibutuhkan pengubahan data analog ke digital dengan kecepatan yang tinggi, maka akan diuraikan hanya ADC dengan kecepatan tinggi (flash ADC).

- FLASH ADC

Konsep yang paling sederhana dan paling cepat dari ADC adalah jenis Paralel Comparator atau disebut Flash ADC. Gambar 2.13 memberikan skema sederhana ADC ini.



Gambar 2.13. Paralel comparator ADC²²⁾

²²⁾ V. Hall, Douglas, Microprocessors and Digital Systems, McGraw-Hill, 1983, hal. 167.

Hubungan antara A1, A2 dan A3 dengan D0 dan D1 dapat dilihat pada tabel 2.8 dibawah ini.

Tabel 2.8. Hubungan keluaran biner dengan keluaran komparator²³⁾

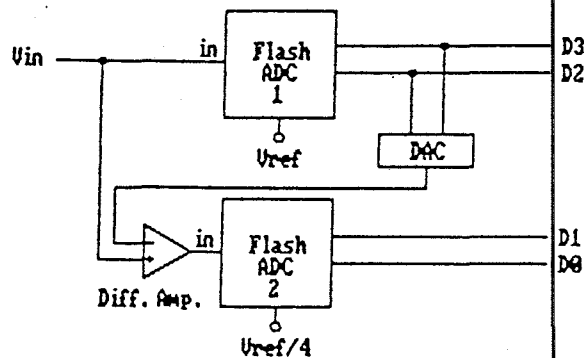
Vin	Output Komparator			Binary output	
	A1	A2	A3	D1	D0
0 - 1/4 Vref	0	0	0	0	0
1/4 Vref - 2/4 Vref	1	0	0	0	1
2/4 Vref - 3/4 Vref	1	1	0	1	0
3/4 Vref - Vref	1	1	1	1	1

Dalam contoh diatas terlihat bahwa untuk 2 bit ADC diperlukan (4-1) ADC. Ini berarti untuk n bit Flash ADC harus digunakan $(2^n - 1)$ komparator. Jumlah komparator yang cukup banyak ini merupakan kelemahan ADC jenis ini.

- HALF-FLASH ADC

Telah diuraikan bahwa Flash ADC memiliki kerugian yaitu jumlah komparator yang banyak. Sehingga dikenal teknik ADC yang disebut Half Flash. Blok diagram ADC 4 bit jenis ini diperlihatkan pada gambar 2.14. pada halaman berikut. Terlihat pada gambar tersebut bahwa di dalamnya digunakan 2 buah Flash ADC 2 bit. Flash ADC yang pertama menghasilkan 2 bit high order yang kemudian dikonversi menjadi tegangan analog lagi oleh sebuah Pengubah Digital ke Analog (DAC) 2 bit.

²³⁾ ibid, hal. 167.



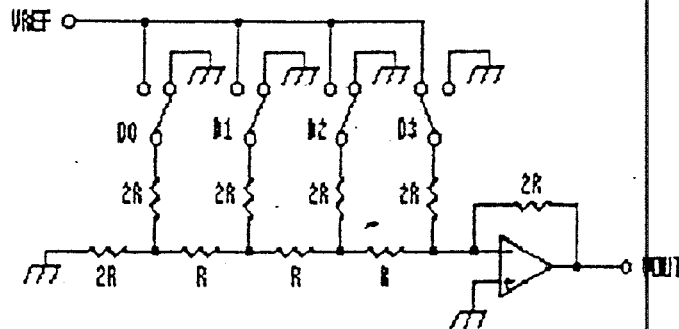
Gambar 2.14. Blok diagram Half Flash ADC

Hasil konversi ini dikurangkan terhadap tegangan input analog dengan sebuah penguat differensial. Hasilnya dikonversi lagi oleh Flash ADC yang kedua. Tetapi dengan tegangan referensi yang berbeda dengan yang pertama. Tegangan referensinya adalah $V_{ref}/4$ atau tegangan pembanding terendah bagi komparator pada Flash ADC pertama. Keluaran Flash ADC kedua, 2 bit, merupakan low order bit.

II.3.2. Konversi Digital ke Analog

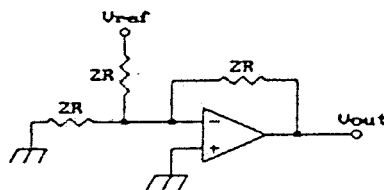
Tidak seperti ADC, DAC yang dipakai kebanyakan dalam bentuk chip dan memakai teknik yang sama dalam mengubah besaran digital menjadi analog. Teknik yang dimaksud adalah R/2R Ladder. Gambar 2.15 memperlihatkan rangkaian dasar R/2R Ladder.

Data D0 sampai D3 menentukan posisi saklar. Dalam gambar tersebut D0 sampai D2 low, sedangkan D3 high, sehingga hanya saklar D3 yang terhubung ke V_{ref} .



Gambar 2.15. Contoh Rangkaian R/2R Ladder DAC²⁴⁾

Oleh sebab itu saklar D0 sampai D2 yang terhubung ke ground membentuk konfigurasi paralel seri sampai ke input inverting dari Penguat Operasional. Jika disederhanakan akan didapat bentuk rangkaian seperti di bawah ini.



Gambar 2.16. Rangkaian pengganti jika D3 on²⁵⁾

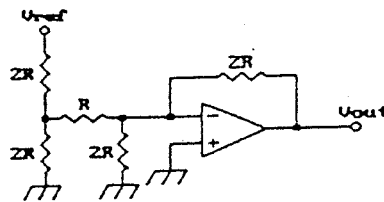
Dari rangkaian tersebut maka keluaran Penguat Operasional yang membentuk inverting amplifier adalah:

$$V_{out} = \frac{V_{ref}}{2} \times 2 \frac{R}{R} = V_{ref}$$

²⁴⁾ ibid, hal. 162.

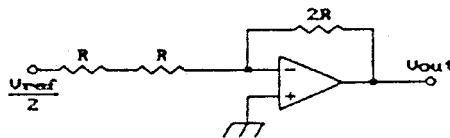
²⁵⁾ ibid, hal. 162.

Sedangkan jika hanya saklar D2 yang on, maka gambar rangkaian penggantinya sebagai berikut.



Gambar 2.17. Rangkaian pengganti jika D2 on²⁶⁾

Rangkaian tersebut dapat lebih disederhanakan lagi menjadi sebagai berikut.



Gambar 2.18. Rangkaian pengganti sederhana jika D2 on²⁷⁾

Dari rangkaian ini maka tegangan keluarannya sekarang adalah:

$$V_{out} = \frac{V_{ref}}{2}$$

Jadi berdasarkan teorema Superposisi tegangan keluaran untuk R/2R Ladder DAC dapat dirumuskan sebagai

²⁶⁾ ibid, hal. 162.

²⁷⁾ ibid, hal. 162.

berikut:

$$V_{out} = V_{ref} \times (D3 + \frac{D2}{2} + \frac{D1}{4} + \frac{D0}{8})$$

Salah satu kerugian penggunaan teknik R/2R Ladder adalah banyaknya resistor yang dipakai, karena tiap bit memakai dua resistor.

II.4. KONTROL PROSES

Untuk melakukan kontrol terhadap suatu proses di bidang industri ada dua cara yang digunakan, kontrol analog dan kontrol digital. Kontrol analog memiliki keunggulan yaitu biaya pembuatan yang lebih murah, sedangkan kontrol digital unggul dalam hal kemudahan perencanaan untuk jenis beban yang berbeda-beda.

II.4.1. Kontrol Analog

Secara umum kontrol analog dibedakan dalam dua jenis, Kontrol Kontinyu dan Kontrol Diskontinyu. Kontrol Diskontinyu umumnya diimplementasikan dalam sistem Ladder, dan dipergunakan pada alat yang tidak terlalu memerlukan keadaan stabil. Kontrol jenis ini memiliki output yang diskontinyu terhadap perubahan error input. Sedangkan kontrol Kontinyu memiliki output yang kontinyu terhadap perubahan error input dan kestabilan yang lebih baik.

- Kontrol Proporsional

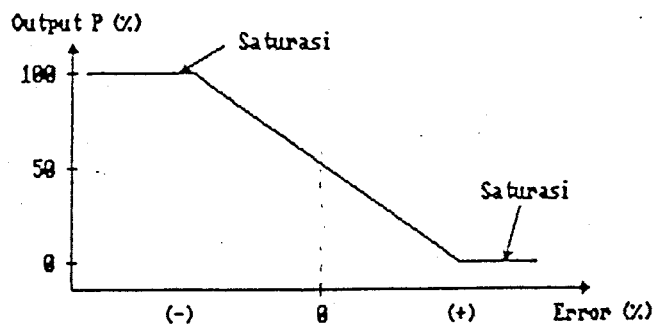
Suatu kontrol proporsional memiliki persamaan sebagai berikut :

$$P = K_p.E_p + P_o$$

Kontrol ini memiliki beberapa karakteristik, yaitu

- Jika error input nol, maka output konstan dan memiliki nilai P_o .
- Perubahan error input 1% mengakibatkan output berubah sebesar $K_p\%$.
- Terjadi error pada keadaan stabilnya yang disebut error offset.

Grafik output terhadap perubahan error untuk kontrol proporsional dengan K_p negatif diperlihatkan pada gambar 2.19 di bawah ini.



Gambar 2.19. Output kontrol Proporsional (K_p negatif)²⁸⁾

²⁸⁾ D. Johnson, Curtis, Process Control Instrumentation Technology, John Wiley & Sons, Inc., 1988 hal. 360.

Karakteristik penting yang dimiliki oleh kontrol jenis ini adalah timbulnya error sisa pada keadaan stabilnya dan dikenal dengan error Offset. Error ini dapat dikurangi dengan memperbesar K_p akan tetapi akan mengakibatkan Band Proporsional semakin sempit, dan semakin mendekati kontrol diskontinu. Besarnya error jenis ini juga sangat dipengaruhi oleh beban yang dikenakan pada output.

- Kontrol Integral

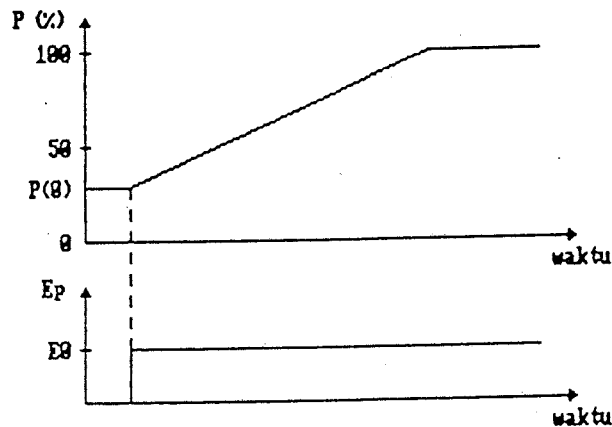
Kontrol jenis ini sering disebut Pengontrol Reset, karena sifatnya yang selalu meniadakan error offset yang timbul pada kontrol Proporsional. Persamaan untuk kontrol Integral ini adalah:

$$\frac{dP}{dt} = K_i E_p$$

Kebalikan dari K_i disebut T_i (periode Integral) dan dinyatakan dalam detik atau menit. Jika persamaan tersebut diintegrasikan pada kedua sisinya akan menjadi:

$$P(t) = K_i \int_0^t E_p(t) dt + P(0)$$

Respon output kontrol integral ini untuk error yang konstan diperlihatkan pada gambar 2.20.



Gambar 2.20. Respon kontrol integral terhadap error yang konstan²⁹⁾

Karakteristik dari kontrol integral ini adalah :

- Jika error input nol, maka output tetap pada kedudukan pada saat error input akan menjadi nol.
- Jika error tidak nol, maka output akan berubah dengan kecepatan $K_i\%$ tiap detik untuk perubahan error 1%.

Walaupun kontrol integral ini memiliki sifat menghilangkan error offset yang timbul pada kontrol proporsional, tetapi dapat juga menimbulkan overshoot yang mengakibatkan sistem memiliki waktu stabil yang lebih lama.

- Kontrol Derivative

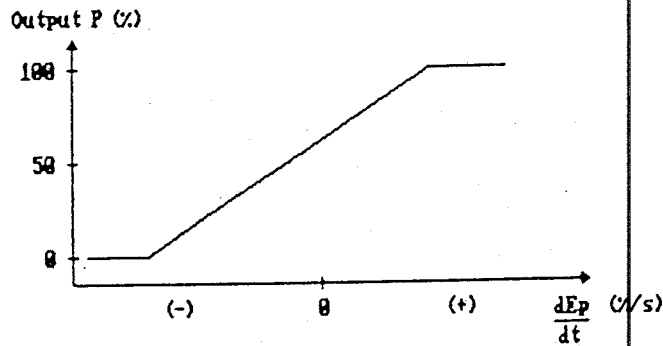
Kontrol jenis ini memiliki output yang bergantung pada kecepatan perubahan error input. Kontrol ini tidak

²⁹⁾ ibid, hal. 364.

dapat berdiri sendiri, karena jika error nol atau konstan maka tidak dihasilkan output. Persamaan kontrol derivative ini adalah :

$$P = K_d \frac{dE_p}{dt}$$

Grafik perubahan output terhadap perubahan kecepatan error diperlihatkan pada gambar 2.21 di bawah ini.



Gambar 2.21. Output kontrol derivative³⁰⁾

Karakteristik kontrol jenis ini adalah :

- Jika error nol atau konstan, tidak ada output.
- Jika error berubah terhadap waktu, output berubah $K_d\%$ untuk perubahan input dengan kecepatan 1% tiap detik.

Kontrol derivative ini memberikan redaman terhadap perubahan output yang cepat. Jika dipadukan dengan kontrol integral dapat dihasilkan jumlah overshoot yang lebih sedikit dan waktu stabil yang lebih cepat.

³⁰⁾ ibid, hal. 367.

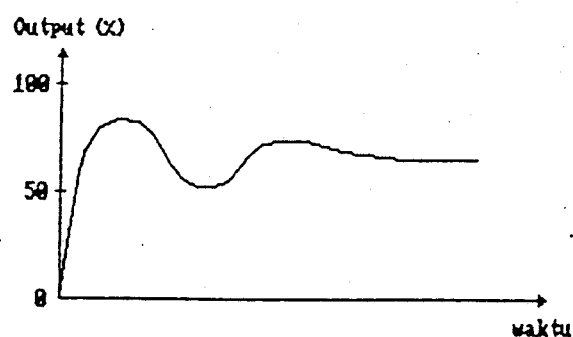
- Kontrol Proporsional-Integral

Kontrol jenis ini adalah kombinasi antara kontrol Proporsional dan kontrol Integral. Persamaannya adalah sebagai berikut :

$$P = K_p \cdot E_p + K_p \cdot K_i \int_0^t E_p dt + G_i(0)$$

Di mana $G_i(0)$ adalah nilai awal suku integral.

Gabungan ini memiliki keunggulan dengan hilangnya error offset jika hanya dipakai kontrol proporsional. Perubahan beban yang sangat mempengaruhi error offset juga ikut ditiadakan, sehingga kedudukan akhir output tidak lagi tergantung pada beban. Grafik output kontrol Proporsional-Integral untuk K_p dan K_i tertentu diperlihatkan pada gambar 2.22.



Gambar 2.22. Output Kontrol Proporsional-Integral³¹⁾

Karakteristik kontrol Proporsional-Integral

³¹⁾ ibid, hal. 370.

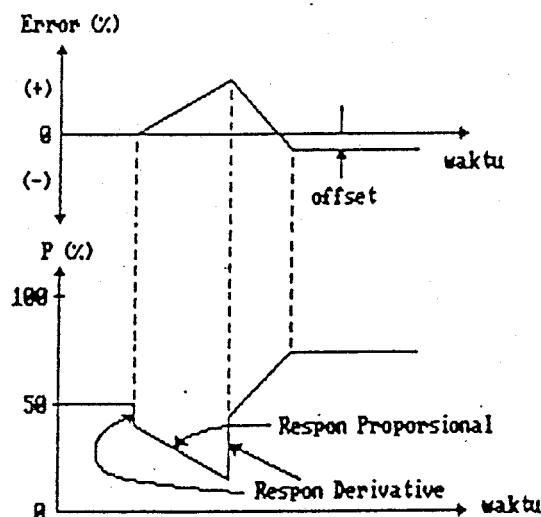
sebagai berikut:

- Jika error bernilai nol, maka output akan bernilai sesuai dengan suku integral saat error menjadi nol.
- Jika error tidak nol, maka suku proporsional menghasilkan output yang sesuai untuk koreksi error tersebut dan suku integral bersifat meniadakan error offset dan pengaruh beban.

Karena sifat-sifatnya tersebut, kontrol jenis ini paling cocok diterapkan pada sistem yang sering mengalami perubahan beban yang cukup besar. Keburukan sistem ini adalah timbulnya overshoot sebelum mencapai keadaan stabilnya.

- Kontrol Proporsional-Derivative

Grafik output kontrol untuk perubahan error diperlihatkan pada gambar 2.23.



Gambar 2.23. Output Kontrol Proporsional-Derivative

Persamaan untuk kontrol ini adalah :

$$P = K_p \cdot E_p + K_p \cdot K_d \frac{dE_p}{dt} + P(0)$$

Kontrol ini dapat dipakai untuk sistem yang memiliki perubahan beban yang cepat. Tetapi kontrol Proporsional-Derivative ini hanya dapat dipakai jika sistem tersebut memiliki toleransi terhadap error offset yang ditimbulkan oleh perubahan beban.

- Kontrol Proporsional-Integral-Derivative

Sistem kontrol yang memiliki banyak keunggulan tetapi cukup rumit adalah PID (Proporsional, Integral, Derivative). Sistem ini lebih unggul dibandingkan kontrol PI (Proporsional-Integral), maupun PD (Proporsional-Derivative). Kekurangan pada kedua kontrol ini diperbaiki pada kontrol PID.

Persamaan kontrol PID dinyatakan sebagai berikut :

$$P = K_p E_p + K_p K_i \int_0^t E_p dt + K_p K_d \frac{dE_p}{dt} + G_i(0) \quad (2-1)$$

Di mana :

P	= Output kontroler
E_p	= Error proses
K_p	= Konstanta proporsional
K_i	= Konstanta integral
K_d	= Konstanta derivative
$G_i(0)$	= Harga awal output kontroler

Dengan penggunaan kontroler ini maka kesalahan offset yang disebabkan oleh kontroler proporsional dapat dihilangkan dengan tetap memberikan respon yang cepat.

II.4.2. Kontrol Digital

Pada penerapan kontrol PID secara digital, maka nilai konstanta K_p , K_i , K_d pada persamaan 2-1 ditentukan secara software. Dan persamaan itu sendiri harus dapat diterapkan secara software. Persamaan 2-1 adalah persamaan kontroller PID untuk waktu kontinyu, sedangkan jika diterapkan dalam komputer yang berperan adalah waktu diskrit. Oleh karena itu persamaan tersebut harus diubah menjadi persamaan differensial diskrit.

Untuk itu mula-mula dilakukan penurunan terhadap kedua sisi persamaan (2-1).

$$\begin{aligned} \frac{dP}{dt} &= K_p K_i \frac{d}{dt} \left(\int E_p dt \right) + K_p K_d \frac{d^2 E_p}{dt^2} \\ \frac{dP}{dt} &= K_p \frac{dE_p}{dt} + K_p K_i E_p + K_p K_d \frac{d}{dt} \left[\frac{dE_p}{dt} \right] \end{aligned} \quad (2-2)$$

Persamaan ini menyatakan berapa banyak perubahan output untuk tiap perubahan input dalam selang waktu yang mendekati nol, yaitu dt . Tetapi di dalam sistem komputer sampling dilakukan untuk tiap interval waktu tertentu, yaitu ΔT . Oleh karena itu persamaan harus diubah dengan menggunakan selang waktu T , dan dE_p

diubah menjadi ΔE_p . Dimana ΔE_p adalah perubahan E_p selama selang waktu T tersebut.

$$\frac{\Delta P}{T} = K_p \frac{\Delta E_p}{T} + K_p K_i E_p + K_p K_d \frac{\Delta}{T} \left[\frac{\Delta E_p}{T} \right] \quad (2-3)$$

Persamaan tersebut dikalikan T pada kedua sisinya. Diperoleh persamaan :

$$\Delta P = K_p \Delta E_p + K_p K_i E_p T + K_p K_d \Delta \left[\frac{\Delta E_p}{T} \right] \quad (2-4)$$

Untuk mengganti ΔP dengan P maka persamaan tersebut harus diubah dengan melihat P sebagai selisih antara nilai yang sekarang dengan nilai yang diambil sebelumnya.

$$\Delta P = P[n] - P[n-1]$$

Hal yang sama juga berlaku untuk input error.

$$\Delta E_p = E_p[n] - E_p[n-1]$$

Dengan demikian persamaan 2-4 dapat ditulis lagi dalam bentuk :

$$P[n] - P[n-1] = K_p (E_p[n] - E_p[n-1]) + K_p K_i E_p T + \frac{K_p K_d}{T} (\Delta E_p[n] - \Delta E_p[n-1]) \quad (2-5)$$

Pada suku yang terakhir terlihat masih dapat

dikembangkan lagi seperti sebelumnya.

$$\begin{aligned}\Delta E_p[n] &= E_p[n] - E_p[n-1] \\ \Delta E_p[n-1] &= E_p[n-1] - E_p[n-2]\end{aligned}$$

Dengan melakukan substitusi ke persamaan (2-4), maka akan didapatkan persamaan :

$$P[n] - P[n-1] = K_p(E_p[n] - E_p[n-1]) + K_p K_i E_p[n] T + \frac{K_p K_d}{T} (E_p[n-1] - 2E_p[n-1] + E_p[n-2]) \quad (2-6)$$

Akhirnya didapatkan persamaan akhir untuk output :

$$P[n] = P[n-1] + K_p(E_p[n] - E_p[n-1]) + K_p K_i E_p[n] T + \frac{K_p K_d}{T} (E_p[n] - 2E_p[n-1] + E_p[n-2]) \quad (2-7)$$

Persamaan ini sudah langsung dapat diprogram dalam bahasa tingkat tinggi seperti C, Pascal, atau Fortran. Tetapi untuk kecepatan yang lebih tinggi dapat digunakan bahasa Assembly dengan memanfaatkan Math Co-Processor dari maksimum sistem yang dibuat dalam tugas akhir ini.

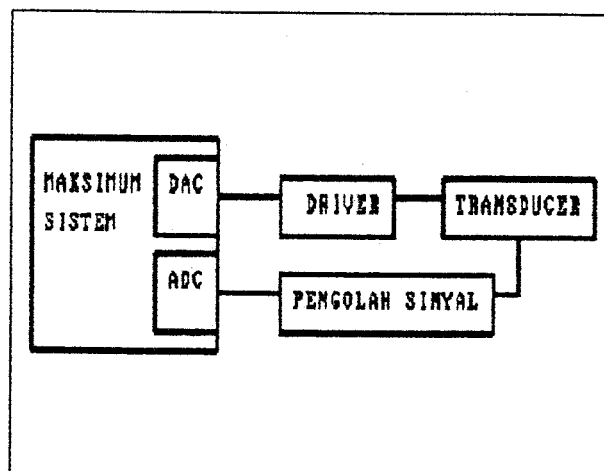
BAB III

PERENCANAAN

III.1. PERENCANAAN HARDWARE

Secara garis besar tugas akhir ini direncanakan atas dua bagian, yaitu : Rangkaian Maksimum Sistem sebagai pemroses utama. Terdiri atas Mikroprosesor 8088 dan Co-prosessor 8087, rangkaian Clock dan rangkaian pembangkit sinyal reset dan wait state, rangkaian pengatur bus, rangkaian buffer data dan latch, rangkaian memori. Bagian kedua merupakan modul aplikasi simulasi PID kontroller. Terdiri atas rangkaian ADC-DAC, rangkaian driver motor dc, transduser, serta pengolah sinyal analog.

Diagram blok peralatan yang direncanakan, secara umum diberikan pada gambar 3.1.



Gambar 3.1 Diagram blok sistem secara umum

III.1.1. Rangkaian CPU, clock dan pembangkit wait state

Rangkaian CPU terdiri atas dua prosessor yang terpasang secara paralel seperti gambar di bawah. Queue Status (QS0 dan QS1) saling dihubungkan agar co-processor dapat menjejaki setiap instruksi yang ditujukan padanya. Untuk instruksi-instruksi tertentu, co-processor dapat melakukan permintaan akses memori langsung (DMA) melalui sinyal dua arah Request/Grant. Untuk memastikan bahwa μP 8088 tidak mengakses memori sebelum co-processor selesai mengeksekusi instruksinya, maka pin Test 8088 dihubungkan dengan pin Busy 8087. Sehingga menghindari pengambilan data yang salah. Jadi μP 8088 akan menunggu pin ini sampai berlogika 0 yang berarti 8087 telah selesai mengeksekusi perintah yang ditujukan kepadanya.

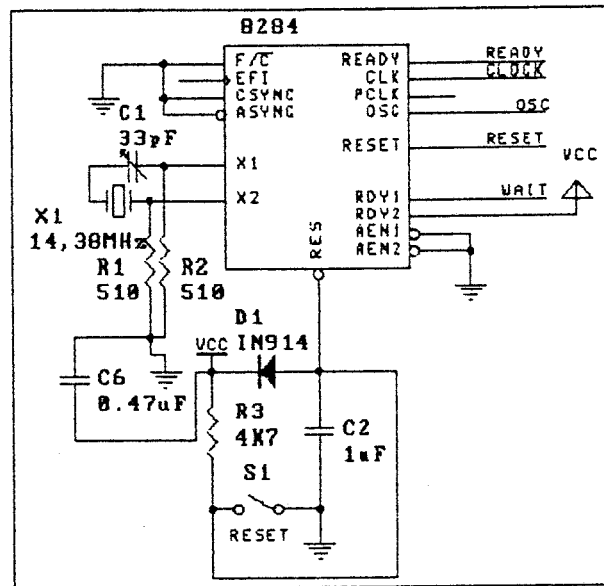
Rangkaian CPU berhubungan secara langsung dengan rangkaian clock. Pembangkit sinyal clock yang dipakai adalah IC 8284. IC ini memiliki 3 keluaran yang memiliki fungsi berbeda, yaitu: pembangkit kontrol reset, pembangkit pulsa-pulsa clock kontinyu, dan pembangkit kontrol sinkronisasi.

- Kontrol reset. RES menerima reset sinyal dari luar. Ketika RES low, sinyal RESET dari flip-flop akan high pada tepi turun clock. Sinyal reset ini dihubungkan dengan reset input μP 8088 untuk mereset μP 8088 secara sinkron. Sinyal reset internal secara langsung

akan dibangkitkan jika catu daya dipasang. Sinyal reset eksternal diperoleh dari rangkaian RC sederhana. Untuk menjamin terjadinya reset maka RESET minimal harus high selama 50 μ s. Dengan mengambil harga R sebesar 4k7 dan harga C sebesar 1 μ F, maka sinyal Reset akan high selama time konstan $t = R.C = 4,7$ ms. Lebih besar dari harga minimal yang dibutuhkan. Diode pada rangkaian tersebut dipakai sebagai jalur pembuangan muatan kapasitor jika catu daya dimatikan.

- **Pembangkit clock.** Ada dua cara untuk membangkitkan sinyal clock. Yaitu secara internal atau eksternal. Cara manapun yang dipakai, frekuensi clock yang masuk ke CPU adalah frekuensi clock yang telah dibagi tiga. Jadi frekuensi clock yang dipakai mestilah 3 kali frekuensi kerja μ P 8088. Pemilihan dilakukan dengan memberi logika 1 atau 0 pada pin input F/C. Direncanakan untuk memakai cara internal sehingga F/C = 0. Agar 8284 membangkitkan clock secara internal, kristal dengan frekuensi 3 kali frekuensi kerja μ P 8088 harus ditambahkan diantara X1 dan X2. Dua buah resistor 510 ohm dipasang terhadap ground untuk stabilitas frekuensi yang dihasilkan. Dan untuk meningkatkan ketelitian, maka diantara X1 dan X2 dipasang sebuah variabel kapasitor. Dengan memasang kristal dengan frekuensi 14,318 Mhz, maka frekuensi kerja μ P 8088 adalah 4,77 Mhz. Gambar 3.2 berikut

menunjukkan rangkaian pembangkit clock μP 8088.

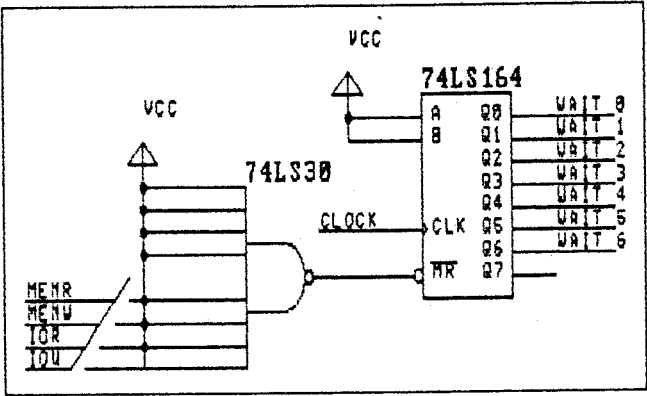


Gambar 3.2. Rangkaian pembangkit clock μP 8088

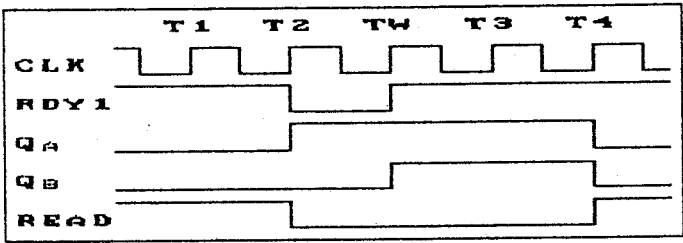
- Pembangkit kontrol sinkronisasi. Sinyal Ready 8284 dipakai untuk membangkitkan wait state untuk μP 8088. Rangkaianannya diperlihatkan pada gambar 3.3(a) di bawah. Inti dari rangkaian tersebut adalah sebuah shift register, IC 74LS164. Dengan keluaran yang bergeser dari QA menuju QF setiap transisi clock dari low ke high. Dengan Serial Input (SI) yang diberi logika high, maka ketika sinyal MEMR low, sinyal QA menjadi high, sedangkan QB masih low (= QA sebelumnya) dan ketika ada transisi clock low ke high berikutnya barulah QB menjadi high. Sehingga terbangkit satu wait state.

Dari gambar rangkaian tersebut tampak bahwa salah

satu dari sinyal-sinyal MEMR/W ataupun IOR/W dapat dipilih untuk membangkitkan wait state serta jumlah wait state dapat dipilih dari satu sampai enam wait state. Misalkan yang dipilih adalah sinyal MEMR dengan satu wait state. Maka kondisi yang terjadi dapat digambarkan dengan diagram waktu gambar 3.3(b).



Gambar 3.3(a). Rangkaian pembangkit sinyal wait state

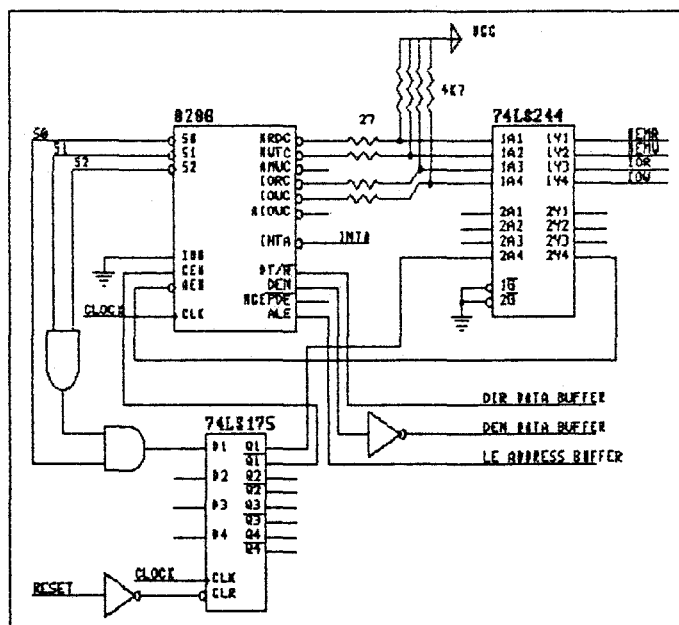


Gambar 3.3(b). Sinkronisasi waktu satu sinyal wait state

III.1.2. Rangkaian Pengontrol Bus

Rangkaian pengontrol bus ini memakai IC 8288. Rangkaian ini membangkitkan sinyal-sinyal MEMR/W, IOR/W, INTA, DT/R, ALE serta DEN. Keluaran MEMR/W dan

IOR/W di-pull up dengan resistor 4K7 untuk menjamin keluaran high pada saat tidak aktif. Rangkaian ini menerima input status S0,S1,S2 dari μ P 8088. Ketiga input ini dikodekan untuk menghasilkan perintah yang bersesuaian. IC 8288 ini dioperasikan pada mode sistem bus dengan memberi level rendah pada input IOB. Sehingga semua sinyal keluaran yang disebutkan di atas diaktifkan. Sedangkan pada kondisi μ P dalam keadaan pasif, input AEN (aktif low) dan input CEN (aktif high) dibuat tidak aktif. Sehingga semua sinyal kontrol tidak aktif. Rangkaian lengkap pengontrol bus diperlihatkan pada gambar 3.4 di bawah ini.



Gambar 3.4. Rangkaian Pengontrol Bus

Untuk menaikkan beban yang mampu dicatu, sinyal-sinyal kontrol ini dibuffer dengan IC 74LS244. Sehingga

arus keluaran logika nol menjadi 24mA dan arus logika satu menjadi 3mA.

III.1.3. Rangkaian Buffer Bus

Rangkaian buffer yang direncanakan terdiri atas IC 74LS373 untuk buffer address dan IC 74LS245 untuk buffer data. Rangkaian buffer address diaktifkan oleh sinyal ALE pada transisi naik. Pada saat transisi naik ALE, address A0-A19 akan dipegang pada keluarannya. Sehingga ketika sinyal read/write aktif, address tetap valid selama siklus tersebut. Input OE diaktifkan (aktif low) jika μP tidak dalam keadaan pasif. Jadi pada kondisi μP aktif OE juga diaktifkan seperti yang dikehendaki. Buffer data memiliki pin DIR yang menentukan arah perpindahan data dari A-B atau B-A serta pin enable perpindahan data tersebut. Pin DIR dihubungkan dengan pin DT/R 8288. Bila terjadi operasi read, maka pin ini akan berlogika nol sehingga terjadi perpindahan data dari B-A dengan syarat DEN aktif. Pada saat operasi write, data berpindah dari A-B. Pin DEN sendiri dihubungkan dengan DEN 8288 sehingga menjamin sinkronisasi antara kedua pin input buffer data tersebut. Jadi kontrol terhadap rangkaian buffer ini sepenuhnya dilakukan oleh rangkaian pengontrol bus.

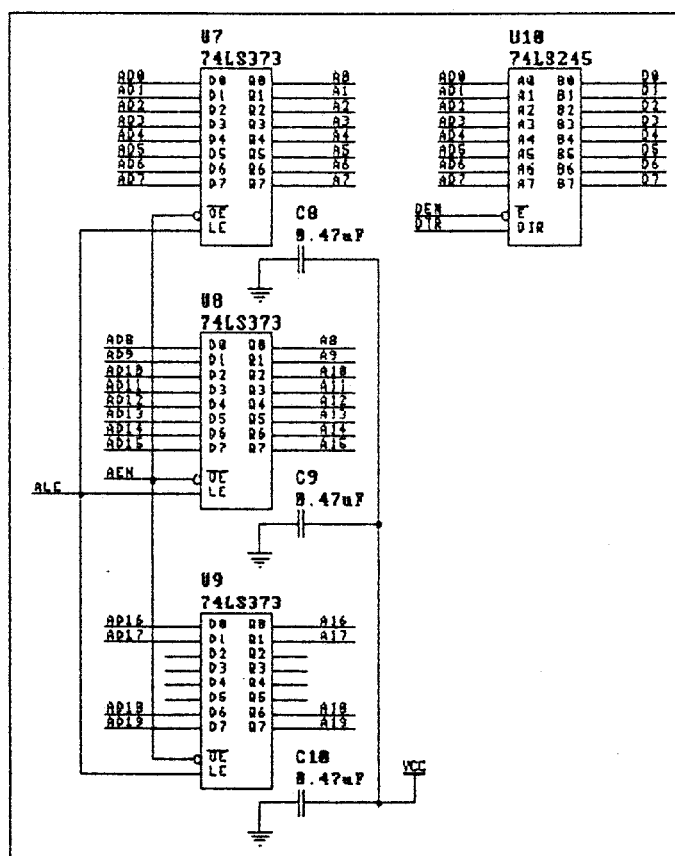
Rangkaian buffer ini juga menaikkan kemampuan pembebanan. Beban yang mampu dicatu address bus untuk logika nol menjadi 24 mA dan untuk logika satu menjadi

2,6 mA. Untuk data bus pada logika nol sebesar 24 mA sedangkan pada logika satu sebesar 3 mA. Fan out dapat dihitung sebagai berikut :

Pada logika nol: $FA_L = 24\text{mA}/1,6\text{mA} = 15$

Pada logika satu: $FA_H = 2,6\text{mA}/40\text{uA} = 65$

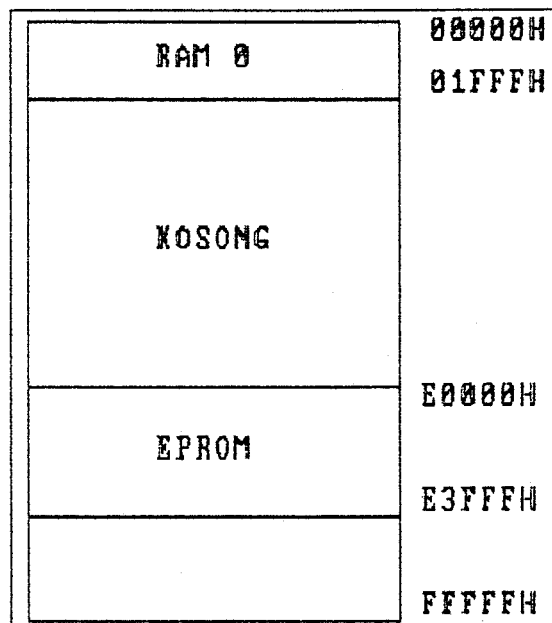
Rangkaian ini juga berfungsi sebagai demultiplekser bagi pin-pin ADO-AD7. Sehingga pada keluarannya, data bus D0-D7 ter-latch pada data buffer sedangkan address bus ter-latch pada buffer address.



Gambar 3.5. Rangkaian buffer data dan address

III.1.4. Rangkaian Interface Memory

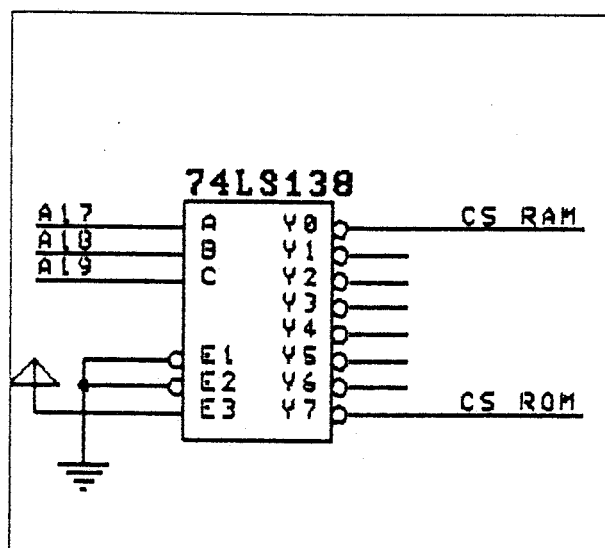
Rangkaian memory yang direncanakan terdiri atas sebuah RAM 6264 (8Kb) dan sebuah EPROM 27128 (16Kb). Memiliki mapping sebagai berikut :



Gambar 3.6. Memory mapping Maximum Sistem 8088

RAM sesuai dengan sifatnya yang mampu ditulisi dan dibaca berulang-ulang, memiliki dua sinyal kontrol yaitu MEMR dan MEMW. Sedangkan ROM hanya bisa dibaca, sehingga hanya dapat diakses melalui MEMR. Kedua sinyal ini dihubungkan dengan pin MEMR dan MEMW dari pengontrol bus setelah melalui buffer 74LS244. Untuk memisahkan RAM yang dialokasikan pada alamat rendah serta ROM pada alamat tinggi dipakai dekoder IC

74LS138. Sebuah dekoder dari tiga ke delapan. Rangkaiannya diperlihatkan pada gambar 3.7 sebagai berikut :



Gambar 3.7. Rangkaian dekoder memori

Pada rangkaian di atas dipakai partial dekoder, dengan hanya mengambil address A17-A19. Sehingga untuk RAM akan terjadi pengulangan address sebagai berikut : 00000-01FFFH, 02000H-03FFFH, 04000H-05FFFH, ..., 1E000H-1FFFFH. Untuk ROM : E0000H-E3FFFH, E4000H-E7FFFH, ..., FC000H-FFFFFH. RAM akan terpilih jika kombinasi input A17-A19 berlogika 0 semua, sedangkan pada kombinasi A17-A19 berlogika 1, ROM akan diakses.

Type RAM yang dipakai adalah 6264ALP-10 dengan waktu akses maksimum 100 ns. Type EPROM yang dipakai

27128A-20 dengan waktu akses maksimum 200 ns. Pada maksimum sistem yang dirancang, waktu akses dapat dihitung seperti berikut :

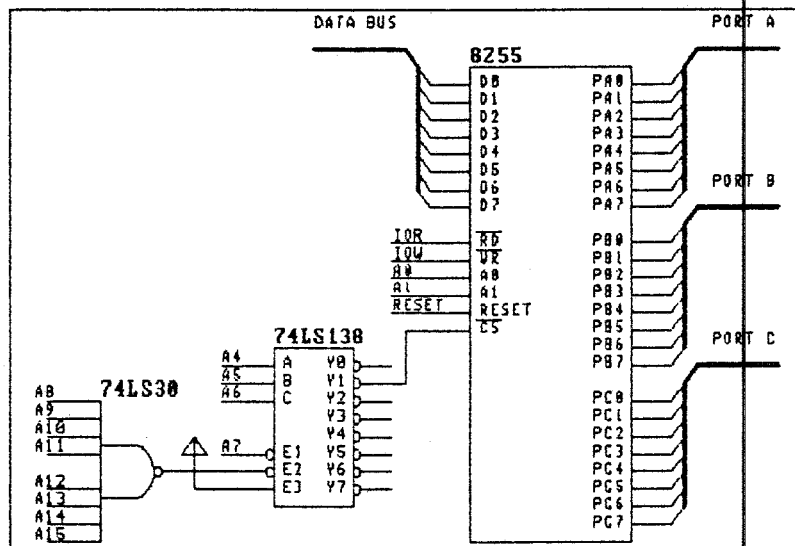
$$\begin{aligned}
 TAVDV &= 3TCLCL - TCLAV_{max} - TDVCL_{min} - \\
 &\quad TDelayMaxGate \\
 &= 3 \times 200ns - 110ns - 30ns - 65ns \\
 &= 395 \text{ ns.}
 \end{aligned}$$

Hasil ini masih lebih besar dari waktu akses RAM maupun EPROM yang dipakai sehingga tidak diperlukan wait state.

III.1.5. Rangkaian Masukan dan Keluaran

Rangkaian ini dipakai untuk menghubungkan maksimum sistem dengan peralatan luar. Berintikan sebuah PPI (Programmable Pheriperal Interface) 8255 yang memiliki 3 alamat port yang dapat diprogram terpisah. Masing-masing port ini terdiri atas 8 pin yang dapat diprogram sebagai input maupun output secara terpisah. Sehingga secara keseluruhan terdapat maksimum 24 titik yang dapat dioperasikan sebagai masukan maupun keluaran. Rangkaian PPI beserta dekodernya diperlihatkan pada gambar 3.8. di bawah ini.

Rangkaian ini dialamati pada alamat memori OFF10H-OFF1FH. Realisasinya dilakukan dengan memakai sebuah NAND gate delapan input untuk address A8-A15 dan sebuah IC dekoder tiga input untuk address A4-A6. Rangkaiannya akan diperlihatkan pada gambar 3.8.



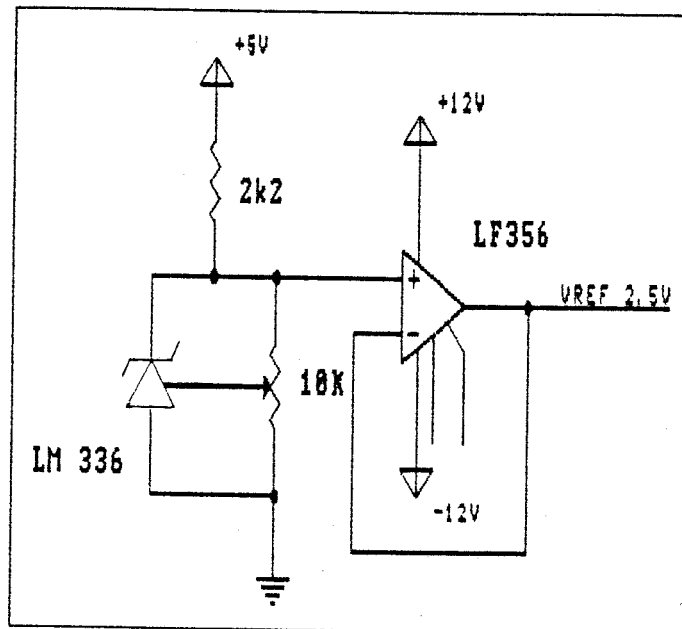
Gambar 3.8. Rangkaian Masukan dan Keluaran

III.1.6. Rangkaian ADC-DAC

Untuk melakukan konversi analog ke digital dan sebaliknya dipakai ADC 0820 dan DAC 0830. ADC 0820 memiliki kecepatan konversi maksimum 2.5 μ s, sedangkan DAC 0830 memiliki waktu keluaran stabil 1 μ s.

Sebagai tegangan referensi ADC dipakai IC LM 336, tegangan referensi presisi. Tegangan referensi ADC sebesar 2.5 volt diperoleh dari rangkaian pada gambar 3.9. Dengan V_{cc} sebesar 5 volt, maka tegangan pada resistor adalah $V_R = 5 - 2.5 = 2.5$ V. Arus yang mengalir ke LM336 memiliki besar yang sama dengan arus yang mengalir pada R (jika tidak dipergunakan potensiometer). Arus ini mempunyai range dari 300 μ A

sampai 10 mA untuk LM336-2.



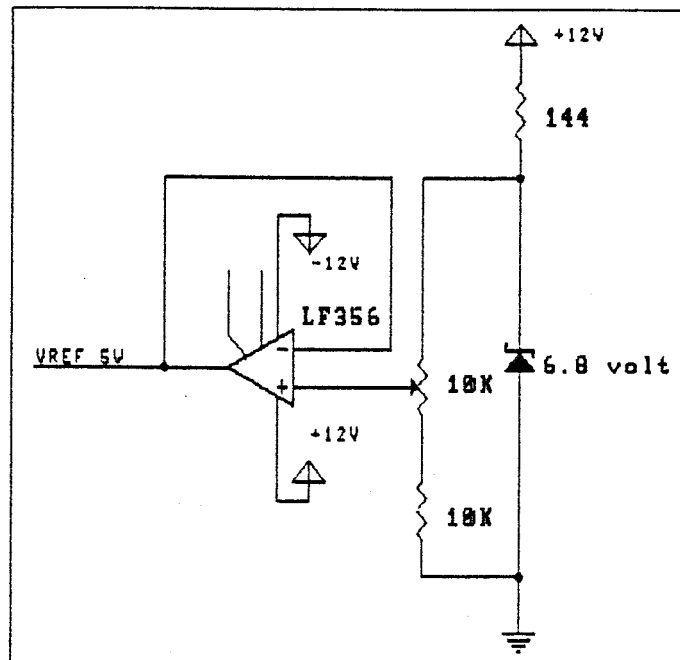
Gambar 3.9. Rangkaian penghasil tegangan referensi ADC

Jika dipilih harga $I = 1 \text{ mA}$, maka R yang dibutuhkan adalah :

$$R = \frac{2.5}{1.10^{-3}} = 2.5 \text{ k}\Omega$$

Dipakai R sebesar $2.2 \text{ k}\Omega$ yang terdapat di pasaran dan arus yang mengalir menjadi 1.14 mA , masih dalam range yang diperbolehkan.

Pada rangkaian DAC dipakai tegangan referensi dari zener diode 6.8 V . Rangkaianannya akan diperlihatkan pada gambar 3.10. Kaki katoda akan menahan tegangan sebesar 6.8 volt . Tegangan ini kemudian diturunkan dengan resistor pembagi tegangan untuk memperoleh tegangan 5V .

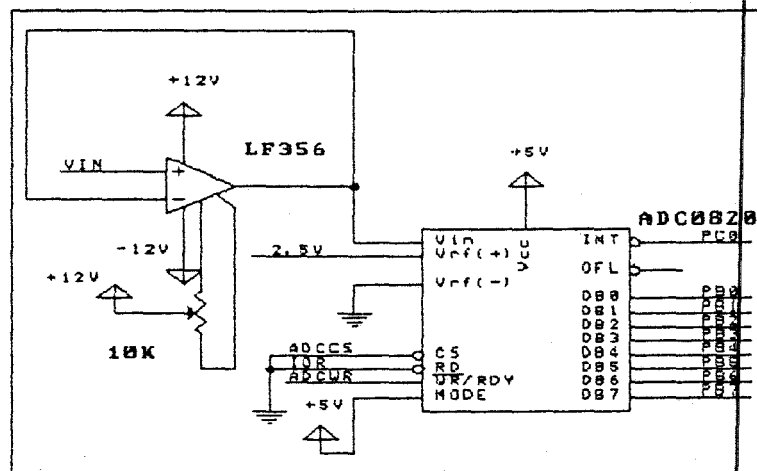


Gambar 3.10. Rangkaian penghasil tegangan referensi DAC

Sebelum diumpankan ke DAC, tegangan ini diberi penguatan arus melalui rangkaian pengikut tegangan dengan OpAmp LF 356.

Rangkaian ADC 0820 dipilih bekerja pada mode WR-RD (pin Mode diberi high) operasi Stand Alone untuk meminimalkan sinyal-sinyal kontrol konversi analog-digital. Pin kontrol yang aktif hanyalah sinyal start konversi (WR/RDY). Sedangkan pin CS dan RD diberi logika low. Start konversi dimulai dengan transisi high ke low sinyal WR/RDY dan minimal harus tetap low selama 600 ns. Akhir konversi ditandai dengan perubahan dari high ke low pada pin INT. Pin ini dihubungkan dengan

Port C bit 0 PPI. Pembacaan data hasil konversi dilakukan melalui Port B. Rangkaian ADC diperlihatkan pada gambar 3.11.

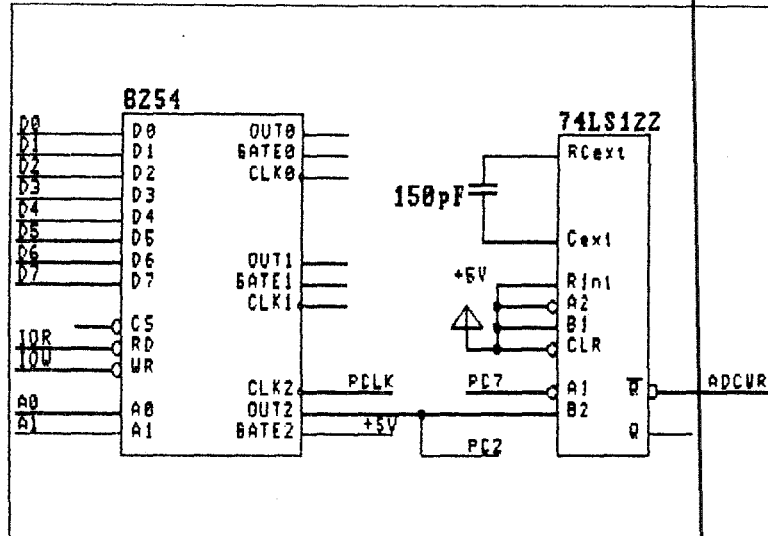


Gambar 3.11. Rangkaian ADC

Sinyal start konversi ADC dihasilkan dengan memakai IC 74LS122, sebuah Monostable multivibrator. Dengan adanya resistor internal pada IC ini maka hanya diperlukan tambahan kapasitor untuk memilih lebar pulsa yang diinginkan. Dengan memilih resistor internal sebesar 10 k Ω maka dari data grafik lebar pulsa terhadap kapasitor luar (Cext) dapat ditentukan besar Cext dengan lebar pulsa yang diinginkan. Gambar 3.12(a) dan 3.12(b) di bawah ini memperlihatkan pembangkit pulsa write ADC serta grafik hubungan lebar pulsa dengan Cext.

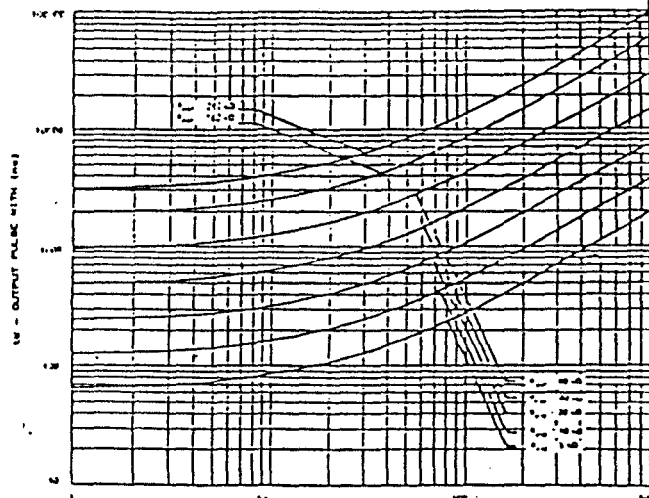
Terlihat bahwa input B2 dihubungkan dengan counter 2 PIT (Programmable Interval Timer) sehingga waktu

sampling ADC dapat diatur melalui perangkat lunak.



Gambar 3.12. (a) Rangkaian penghasil pulsa write ADC

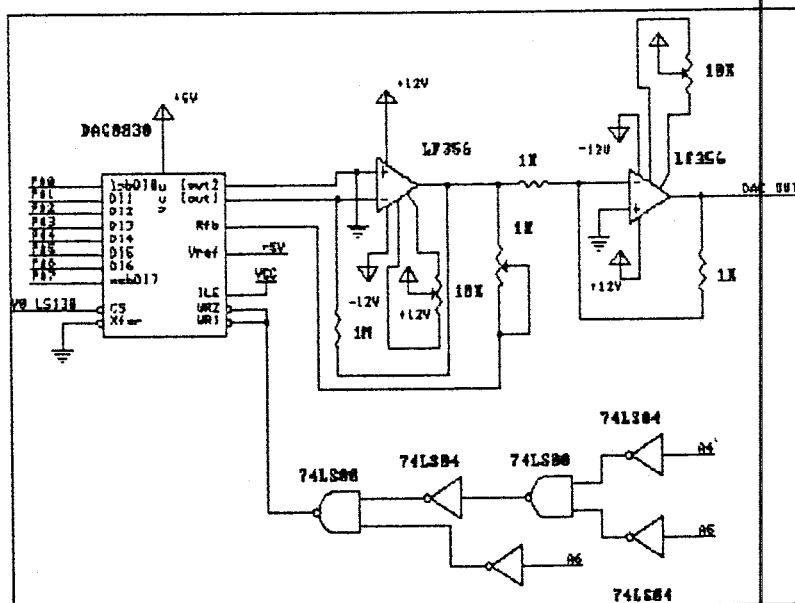
Sedangkan input A1 dihubungkan dengan port C7 untuk mengenable start konversi ADC. Counter 2 PIT tersebut dapat didisable/enable secara manual dengan rangkaian flip-flop sederhana memakai IC 74LS00.



Gambar 3.12(b). Grafik lebar pulsa (tw) terhadap Cext

Dari grafik tersebut jika dipilih $R_{ext} = 10 \text{ k}\Omega$ (sama dengan resistor internal) dan t_w (lebar pulsa) = 700 ns, maka diperoleh C_{ext} sebesar 150 pF. Sedangkan untuk menggunakan resistor internal, maka pin Rin dihubungkan ke V_{cc} .

Sedangkan DAC 0830 memerlukan rangkaian yang lebih kompleks, karena keluarannya berupa arus yang harus diubah dulu menjadi tegangan. Kontrol terhadap DAC diperoleh dari output Y0 dekoder serta kombinasi address A4-A6. Jadi pada saat perintah konversi diberikan, pin CS (yang terhubung ke IOW) serta pin WR1 dan WR2 (yang dikontrol oleh A4-A6) harus berlogika low. Dengan menghubungkan port A ke pin data D0-D7 maka start konversi dapat diberikan secara software.



Gambar 3.13. Rangkaian DAC

Untuk mengubah keluaran arus ke besaran tegangan dipakai OpAmp LF 356. Agar skala penuh DAC dapat diatur maka pada OpAmp pertama diberikan R2 variabel sebesar 1 k Ω . Sedangkan tahanan R1 sebesar 1 M Ω berfungsi untuk menurunkan sedikit skala penuh. Dengan R1 ini maka skala penuh dapat diatur melalui R2. Untuk pengaturan harga nol diberikan potensiometer sebesar 10 k Ω pada offset OpAmp.

Tegangan keluaran dari OpAmp yang pertama ini adalah:

$$V_{out} = - \frac{V_{ref}(DigitalInput)}{256}$$

Agar tegangan dapat memiliki polaritas positif, maka tegangan output diberi inverting amplifier dengan penguatan satu. Hasil akhir penguatan OpAmp kedua menjadi :

$$V_{out} = \frac{V_{ref}(Digital input)}{256}$$

Hasilnya adalah tegangan keluaran yang bervariasi dari 0 volt sampai +(Vref - 1LSB).

III.1.7. Pengolah sinyal analog

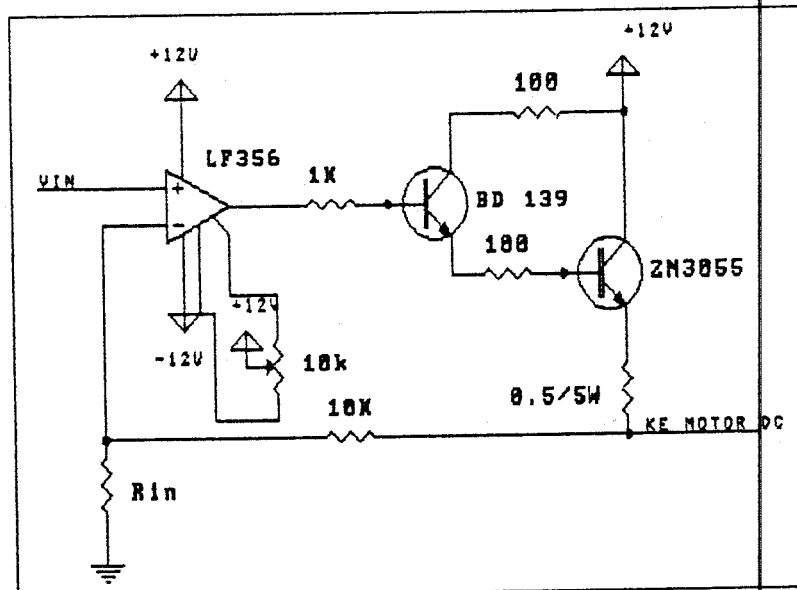
Rangkaian pengolah sinyal ini terdiri atas rangkaian driver motor DC, filter low pass.

- Rangkaian driver motor DC

Rangkaian ini direncanakan untuk memperkuat tegangan positif keluaran DAC. Direncanakan tegangan input motor maksimum 10 volt (dibatasi oleh Vsat

OpAmp). Rangkaian driver ini diperlihatkan pada gambar 3.14.

Rangkaian di bawah pada dasarnya adalah sebuah penguat tidak membalik. Untuk memperkuat arus keluaran OpAmp, ditambahkan dua buah transistor dalam konfigurasi Darlington. Tegangan keluaran dibatasi oleh tegangan saturasi OpAmp karena letak transistor pada loop umpan balik. Dengan catu daya pada OpAmp sebesar ± 12 volt maka tegangan input berkisar dari $+V_{sat}$ sampai $-V_{sat}$, yaitu ± 10 volt. Dengan penguatan dua kali, maka tegangan keluaran DAC dari 0 sampai 5 volt menjadi 0 sampai 10 volt yang sudah cukup untuk menggerakkan motor.

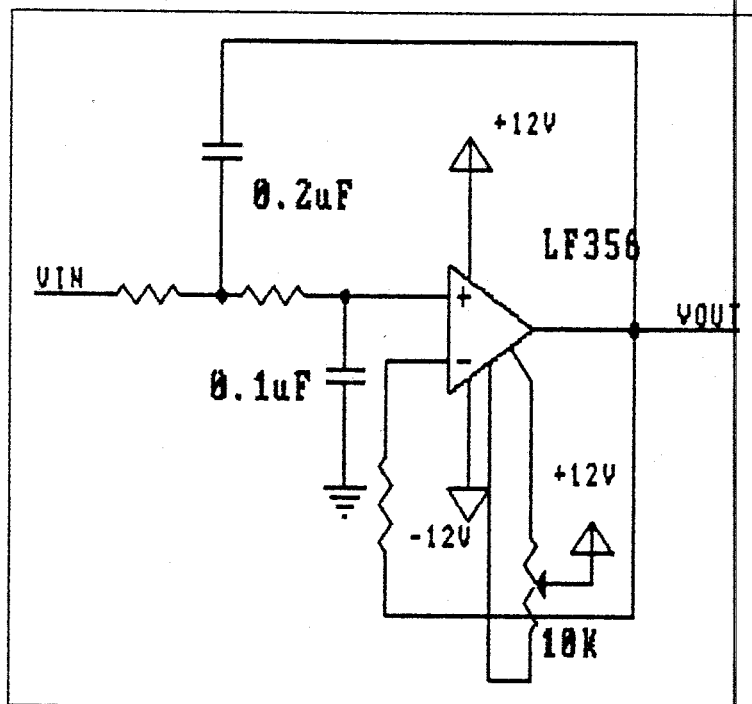


Gambar 3.14. Rangkaian driver motor DC

- Rangkaian filter low pass

Tegangan emf motor tacho masih mengandung komponen frekuensi tinggi. Oleh karenanya, perlu ditambahkan filter low pass sebelum tegangan ini diinputkan pada ADC. Frekuensi cut-off ditentukan pada tegangan input maksimum motor. Yaitu sekitar 70 Hz. Rangkaian low pass yang direncanakan adalah Butterworth -40 dB/dekade. Dengan $R1 = R2 = 100 \text{ k}\Omega$, dan $R3 = 200 \text{ k}\Omega$, nilai kapasitor dihitung dari persamaan :

$$C = \frac{0.707}{\omega_c R} = \frac{0.707}{50 \text{ } 100\text{k}\Omega} = 0.14\mu\text{F}$$



Gambar 3.15. Rangkaian Filter low pass

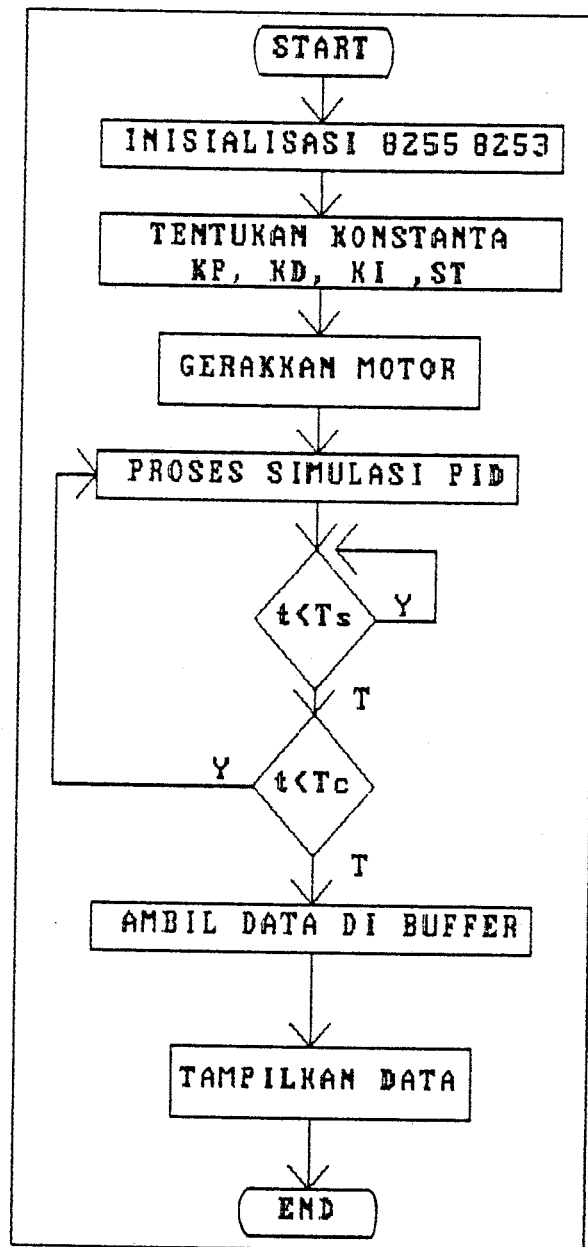
III.2. PERENCANAAN PERANGKAT LUNAK

Perangkat lunak yang dibuat mempergunakan bahasa Assembly. Hal ini disebabkan karena kemudahan pemakaian dalam mengakses perangkat keras serta kecepatan proses yang sangat diperlukan pada simulasi PID yang direncanakan.

Karena adanya bagian derivative maka waktu pengambilan data harus selalu tetap. Karena adanya proses perhitungan yang mengikuti proses pengambilan data, maka diperlukan suatu penghitung waktu. Untuk keperluan itu dipakai PIT 8253 pada operasi Mode 3 (square wave generator). Setelah diinisialisasi, akan segera dihasilkan pulsa dengan periode yang tetap. Inisialisasi PIT yang dipilih adalah counter 2, mode 3, 16 bit binary count LSB lalu MSB. Begitu program tersebut dijalankan, PIT segera mengeluarkan pulsa kotak. Untuk mengetahui berakhirnya suatu periode dipergunakan program berikut :

```
@cek:      mov dx,PC_dac
            in al,dx
            test al,100b
            jnz @cek
@cek1:
            in al,dx
            test al,100b
            jz @cek1
```

Diagram alur program utama digambarkan sebagai berikut:



Gambar 3.17. Diagram alur program utama

Program tersebut diatas akan menunggu terjadinya transisi dari low ke high pada Port C2 , sebagai awal proses simulasi berikutnya.

- Proses simulasi PID

Seperti telah diuraikan pada bab Teori penunjang, maka rumus yang dapat diterapkan ke bahasa Assembly dengan mudah didapat dari persamaan (2-9) sebagai berikut:

$$P[n] = P[n-1] + K_p(E_p[n] - E_p[n-1]) + K_p K_i E_p[n] T + \frac{K_p K_d}{T} (\Delta E_p[n] - \Delta E_p[n-1])$$

Dimana E_p adalah error, yaitu selisih antara Set Point dengan Variabel Proses, $E_p = ST - PV$. Untuk mempermudah dan mempercepat perhitungan dari rumus diatas, nilai-nilai yang tidak berubah selama simulasi dihitung dulu, sehingga rumus tersebut menjadi :

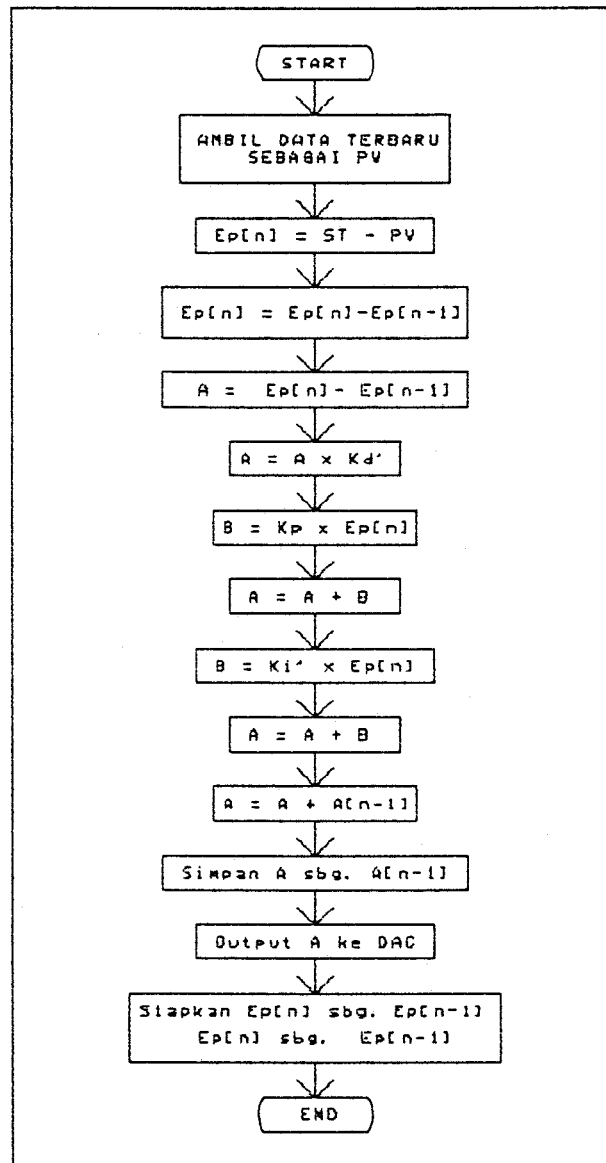
$$P[n] = P[n-1] + K_p(E_p[n] - E_p[n-1]) + K_i' E_p[n] + K_d' (\Delta E_p[n] - \Delta E_p[n-1])$$

di mana :

$$K_i' = K_p K_i T$$

$$K_d' = \frac{K_p K_d}{T}$$

Diagram alir proses simulasi PID adalah sebagai berikut :



Gambar 3.18. Diagram alur proses simulasi PID

Diagram alur di atas disesuaikan dengan struktur pemrograman math coprocessor 8087. Diagram di atas hanya memperlihatkan proses simulasi satu tahap. Untuk tahap simulasi berikutnya diperlukan pemeriksaan akhir untuk memberikan selang waktu yang tetap .

Dalam implementasinya ke bahasa assembly untuk math coprocessor 8087, maka semua variabel dalam diagram alur di atas adalah delapan stack dari 8087. Oleh sebab itu penggunaan stack tersebut harus betul-betul hemat.

Agar kecepatan betul-betul maksimum, semua nilai konstanta, yaitu K_p , K_i' , K_d' dan ST disimpan terlebih dahulu ke stack. Harga awal $Ep[n]$ dan $\Delta Ep[n]$ adalah nol. Program inisialisasi ini adalah :

```
finit
fld dword ptr[Kd']
fld dword ptr[Ki']
fld dword ptr[Kp]
fld dword ptr[ST]
fldz
fldz
fldz
```

Sedangkan diagram alir gambar 3.18 dapat diimplementasikan sebagai berikut :

```
fild word ptr[PV]

fsubr st,st(4)
fst st(3)

fsubrp st(2),st
fsubr st,st(1)

fld st(6)
fmulp st(1),st

fld st(4)
fmul st,st(2)

faddp st(1),st

fld st(5)
fmul st,st(3)
```

```
faddp st(1),st

fiadd word ptr[di+bx]
inc bx
inc bx
fistp word ptr[di+bx]

mov dx,PA_dac
mov al,byte ptr[di+bx]
out dx,al

fxch st(1)
fld st(1)
```

BAB IV

PEMBUATAN DAN UJI COBA

Untuk memudahkan perbaikan jika terjadi kesalahan, maka peralatan yang dibuat dibagi menjadi empat modul. Modul utamanya adalah maksimum sistem yang dibuat, modul ADC-DAC, dan modul Display beserta Keypad. Pada modul maksimum sistem terdapat sebuah slot untuk memasang modul interface ADC-DAC.

4.1. Pengujian Modul Maksimum Sistem

Pengujian modul maksimum ini meliputi pengujian terhadap masing-masing bagian yaitu : rangkaian penghasil clock, rangkaian pengontrol bus, rangkaian buffer data dan address. Semua pengujian ini dilakukan pada kondisi steadynya. Sedangkan pengujian terhadap timing problem tidak dapat dilakukan karena diperlukan adanya sebuah logik analyzer.

- Pengujian rangkaian clock

Pengujian dilakukan dengan melihat langsung keluaran sinyal Clk 8284 pada oscilloscope. Sinyal reset diamati dengan melihat perubahan dari high (tidak aktif) ke low (aktif) ketika saklar push button ditekan. Sinyal ready diamati dengan memberi logika low pada pin input RDY1. Keluaran sinyal ready berlogika low saat logika low diinputkan pada RDY1 sesuai dengan yang diharapkan.

-Pengujian rangkaian pengontrol bus

Keluaran sinyal-sinyal kontrol tergantung pada kondisi input S0, S1, dan S2. Sehingga pengujian dapat dilakukan dengan memberi kombinasi logik pada ketiga input tersebut lalu diamati sinyal kontrol yang bersesuaian. Pengujian dilakukan dalam keadaan input CEN dan AEN aktif.

- Pengujian rangkaian buffer address

Untuk pengujian rangkaian buffer ini maka pada pin ADO-AD19 up 8088 diberi kombinasi logika tertentu. Keluaran AO-A19 diamati setelah sinyal latch enable diberikan pada input LE IC 74LS373. Pin AO-A19 akan sama dengan kombinasi pada pin ADO-AD19 jika rangkaian buffer berfungsi dengan benar. Hal yang sama dilakukan pada data buffer. Tetapi sekarang sinyal input DEN diberi logika rendah dan sinyal input DTR diberi logika tinggi. Kemudian diamati kondisi logika pada sisi keluaran. Logika pada sisi input ADO-AD7 harus sama dengan keluaran DO-D7 jika buffer bekerja dengan benar. Dari hasil pengujian dengan beberapa kombinasi diperoleh bahwa input selalu sama dengan hasil pada output.

4.2. Pengujian Modul ADC-DAC

Sebagai uji coba dilakukan pembacaan data ADC setelah bagian pengolah sinyal analog dikalibrasi. Dengan memberi tegangan input tertentu kemudian

konversi dilakukan berulang-ulang sebanyak 5 kali.

Hasil pembacaan data diberikan dalam tabel 4.1 di bawah ini.

Tabel 4.1 Hasil pengukuran ADC

INPUT ANALOG (VOLT)	OUTPUT DIGITAL				
0.0	25	25	24	24	25
0.5	56	56	56	56	57
1.0	110	111	110	111	111
1.5	158	158	158	158	159
2.0	207	207	207	207	207
2.5	255	255	255	255	255

Dari data yang telah diambil tersebut dapat dicari rata-rata dan deviasi untuk tiap titik tegangan tersebut. Hasilnya diberikan pada tabel 4.3 di bawah ini.

Pengukuran DAC dilakukan dengan memberikan input digital dari 0 sampai 255. Hasil pengukuran diperlihatkan pada tabel 4.2 pada halaman berikutnya.

Tabel 4.2. Hasil Pengukuran DAC

INPUT DIGITAL	OUTPUT ANALOG(V)
0	0.0
30	0.6
60	1.18
90	1.77
120	2.35
150	2.95
180	3.53
210	4.12
240	4.71
255	5.00

Secara teoritis, dengan mempergunakan persamaan pada bab perencanaan, diperoleh kesalahan pengukuran yang berkisar dari 10 sampai dengan 20 milivolt.

Perbedaan ini disebabkan antara lain oleh kesalahan offset pada masing-masing OpAmp yang dipakai. Dimana keluarannya tidak benar-benar nol pada saat input masih bertegangan nol.



Tabel 4.3. Statistika data ADC

Input analog (volt)	Digital output		
	Rata-rata	Deviasi standar	
0.0	24.6		0.55
0.5	56.2		0.48
1.0	110.6		0.55
1.5	158.2		0.45
2.0	207		0.00
2.5	255		0.00

Hasil statistika di atas menunjukkan bahwa kebanyakan tegangan memiliki error sedikit di atas 1 LSB walaupun ada yang lebih rendah. Kesalahan ini pada ADC 0820CCN menurut data book adalah ± 1 LSB. Ini jika ADC dipakai tidak pada kecepatan maksimumnya. Hal ini disebabkan oleh semakin besarnya error dengan semakin sempitnya pulsa Start Conversion yang diberikan. Dan juga tergantung pada jarak pulsa pembacaan dari pulsa Start Conversion.

Jika dicari regresi linier untuk nilai output rata-rata terhadap tegangan input, hasilnya adalah :

$$Y = 30.69 + 83.66$$

Hasil tersebut menunjukkan bahwa tiap kenaikan 1 volt, ada kenaikan output digital sebesar 83.66. Atau kenaikan 1 bit disebabkan oleh kenaikan tegangan 11.95 milivolt.

Untuk mengetahui pengaruh kombinasi konstanta Proporsional, Integral, dan Derivative dilakukan analisa dengan memberikan kombinasi konstanta yang berbeda-beda. Lalu diamati grafik perubahan tegangan yang timbul pada motor tacho. Pada halaman berikut ditampilkan berturut-turut kombinasi-kombinasi K_p , K_d dan K_i sebagai berikut :

1. $K_p = 1$, $K_d = 0$, $K_i = 0$, $ST = 75 \%$
2. $K_p = 1$, $K_d = 0$, $K_i = 1$, $ST = 75 \%$
3. $K_p = 1$, $K_d = 0$, $K_p = 15$, $ST = 75 \%$

Dari ketiga grafik tersebut tampak bahwa kenaikan K_i mengurangi error steady state. Tetapi menimbulkan overshoot pada awal gerakan motor karena sentakan pada saat motor mulai bergerak.

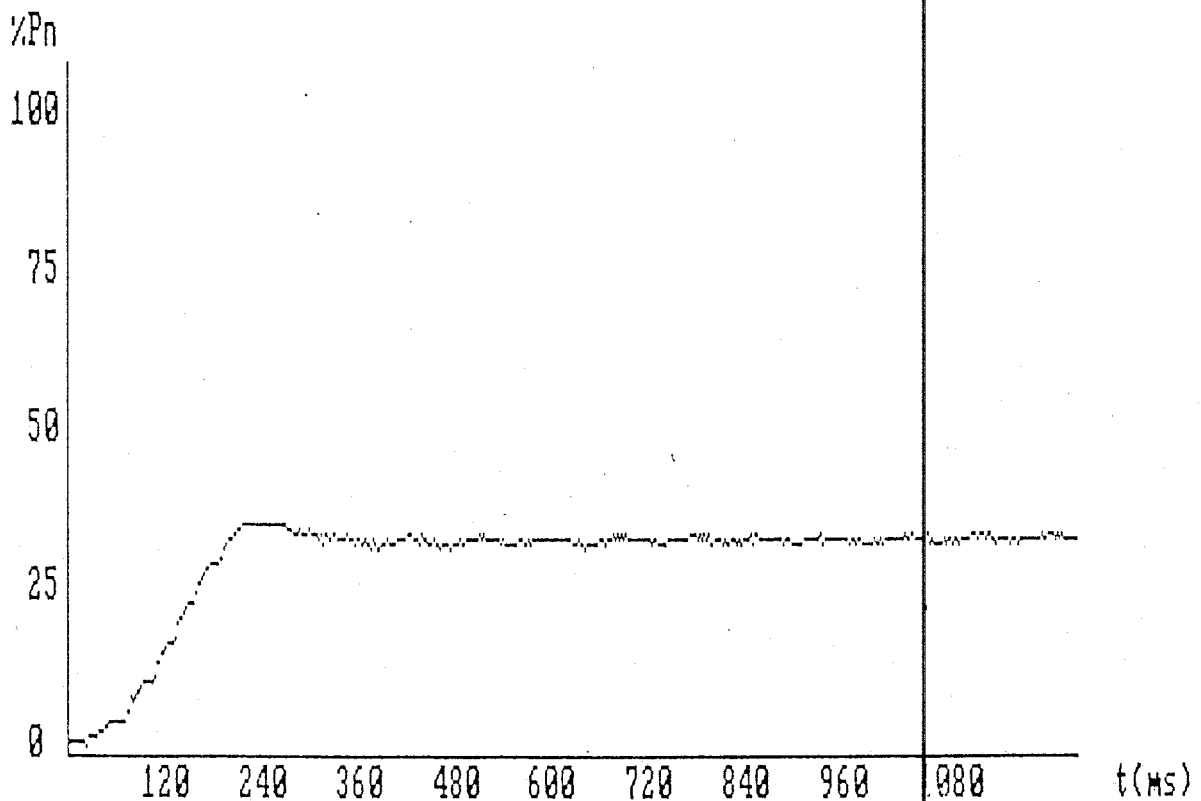
Konstanta-konstanta yang sama diulangi untuk $ST = 50\%$ pada tiga grafik berikutnya. Terlihat bahwa terjadi kemiripan di mana kenaikan K_i mengurangi error steady state namun mengakibatkan overshoot jika nilainya dinaikkan cukup besar.

Pengaruh K_d diamati pada dua grafik berikutnya sebagai berikut:

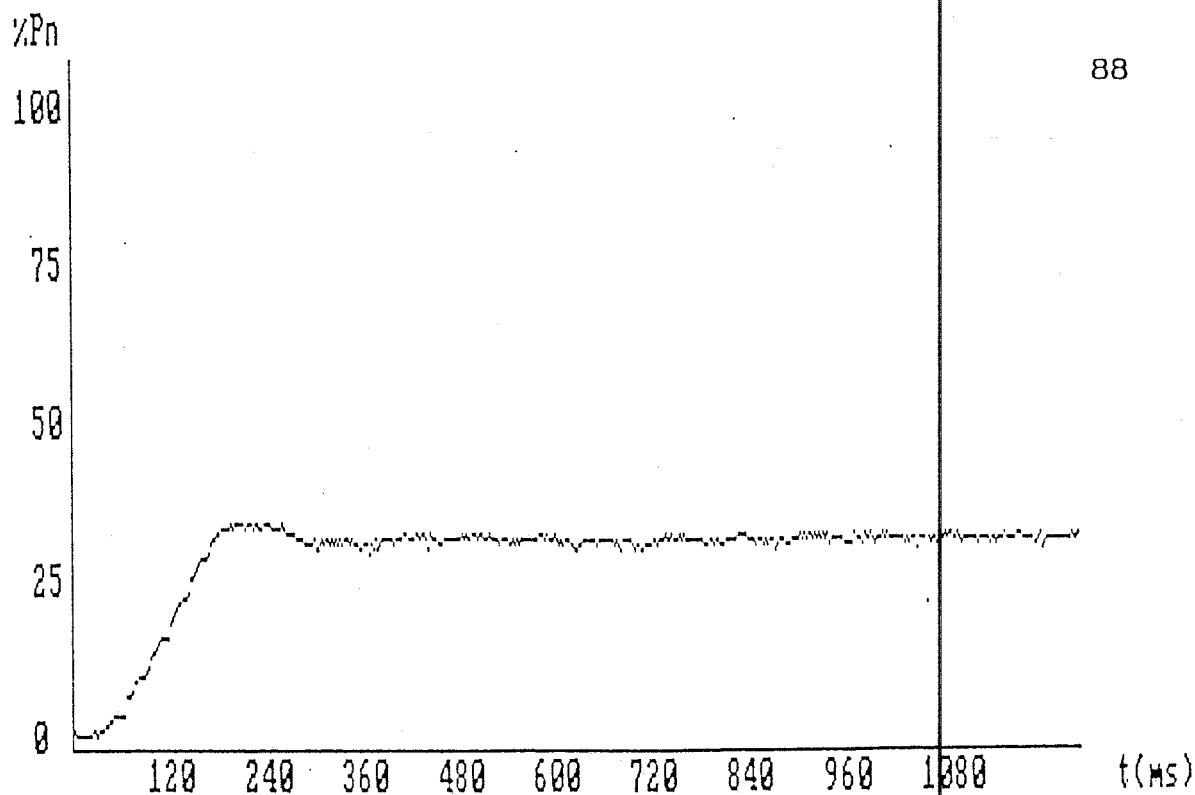
$$K_p = 1 \text{ , } K_d = 0.03 \text{ , } K_i = 1 \text{ , } ST = 50\%$$

$K_p = 1$, $K_d = 0.5$, $K_i = 1$, $ST = 50\%$

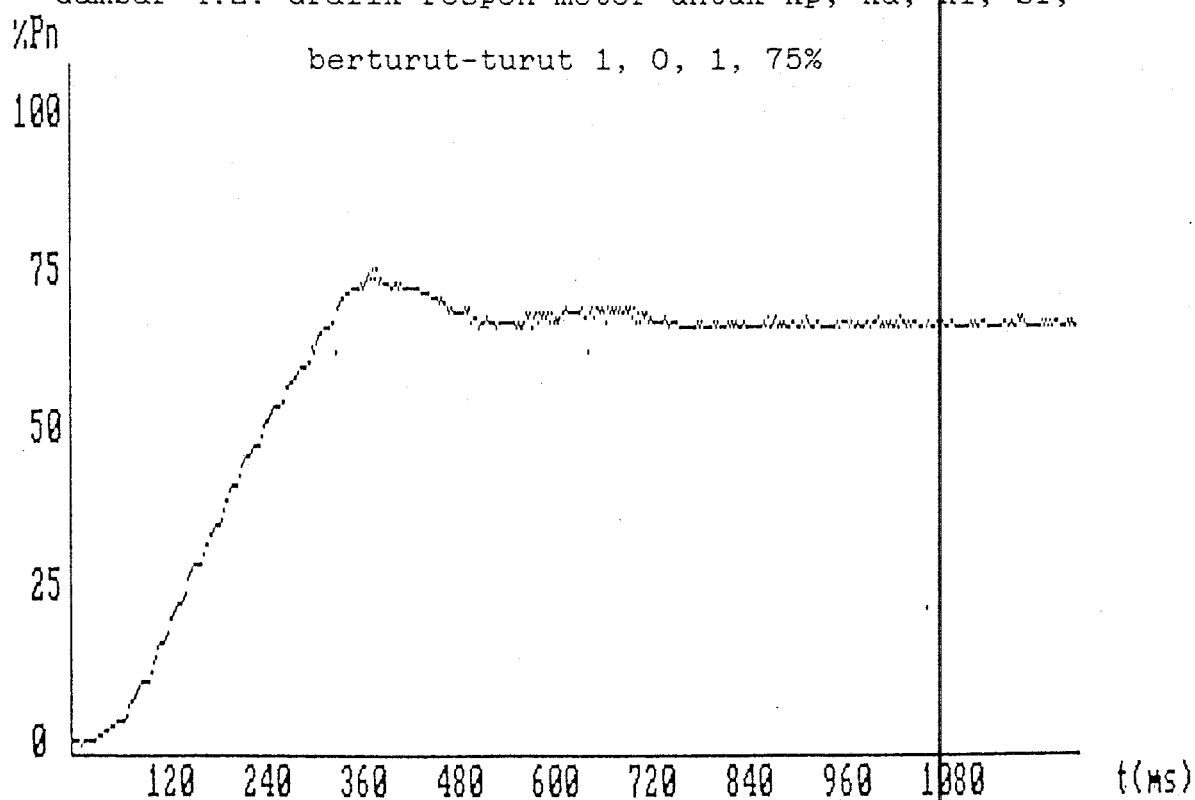
Pada $K_d = 0.03$ didapatkan respon yang bagus, sedangkan jika K_d dinaikkan terjadi sentakan-sentakan saat motor mulai bergerak.



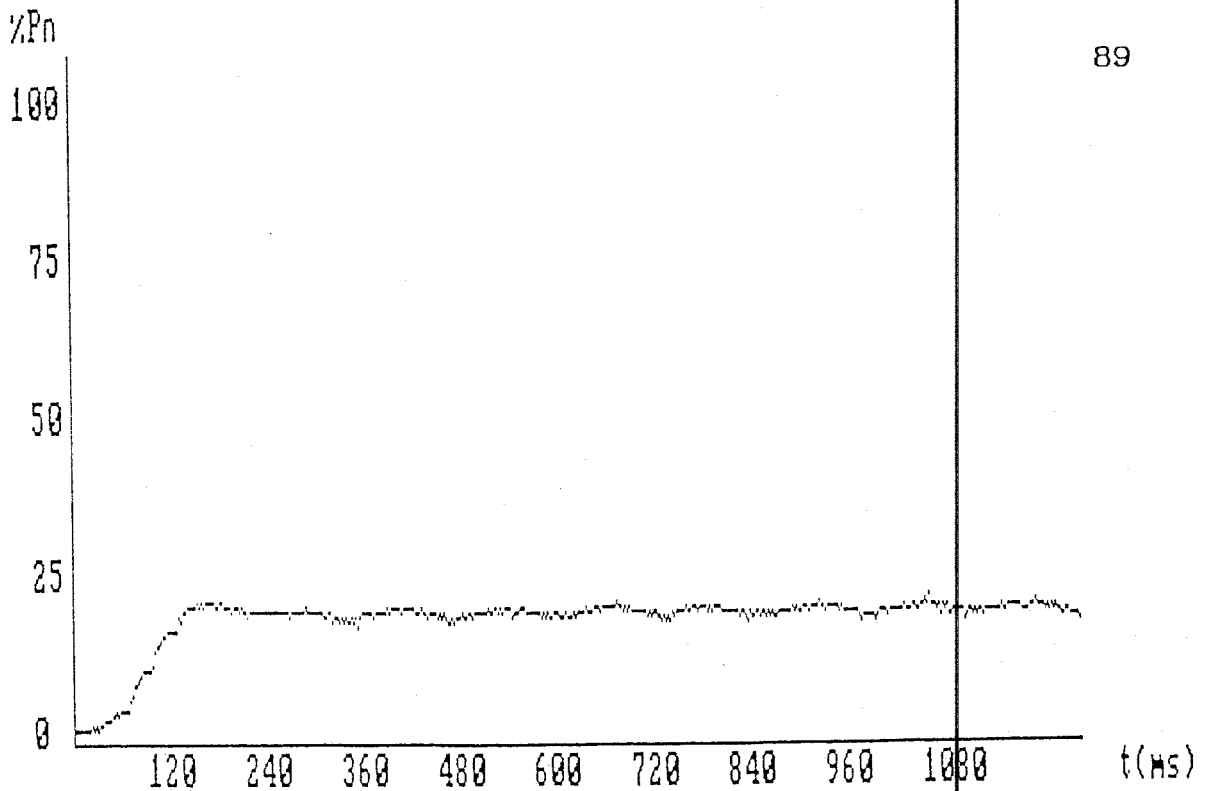
Gambar 4.1. Grafik respon motor untuk K_p , K_d , K_i , ST , berturut-turut 1, 0, 0, 75%



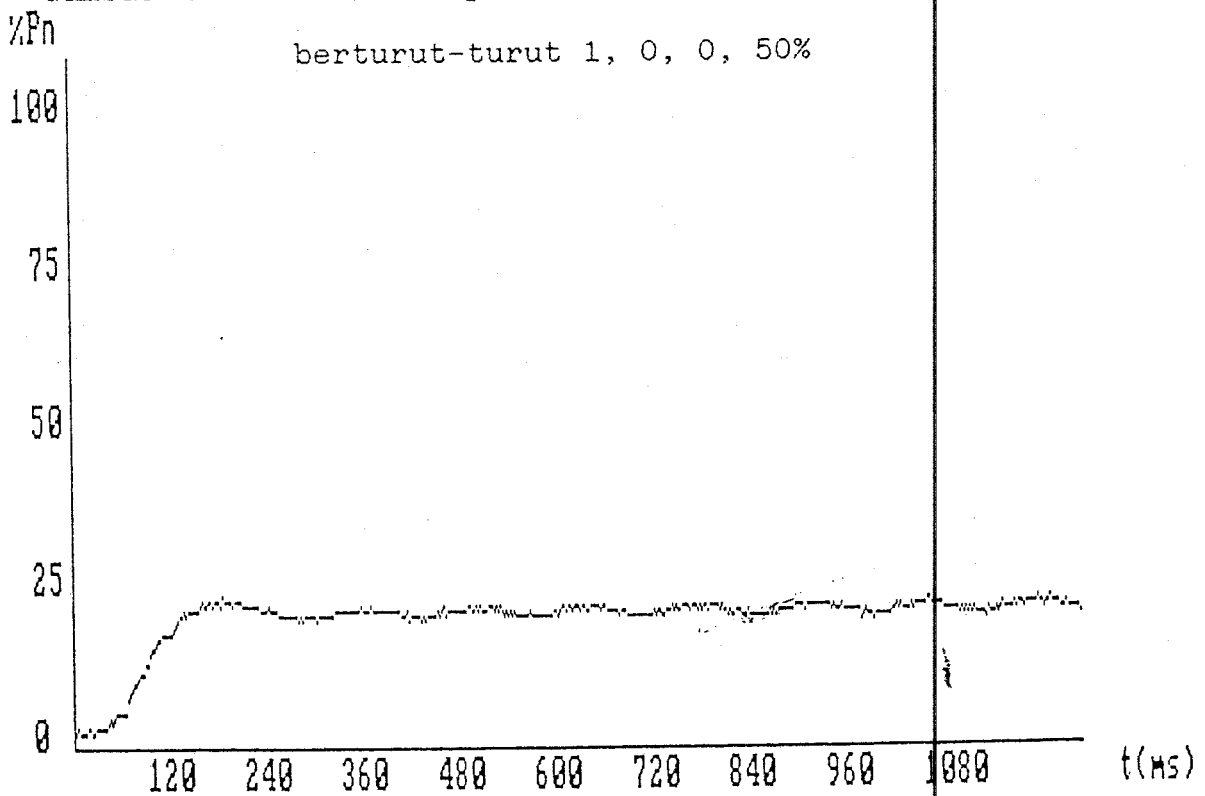
Gambar 4.2. Grafik respon motor untuk K_p , K_d , K_i , ST , berturut-turut 1, 0, 1, 75%



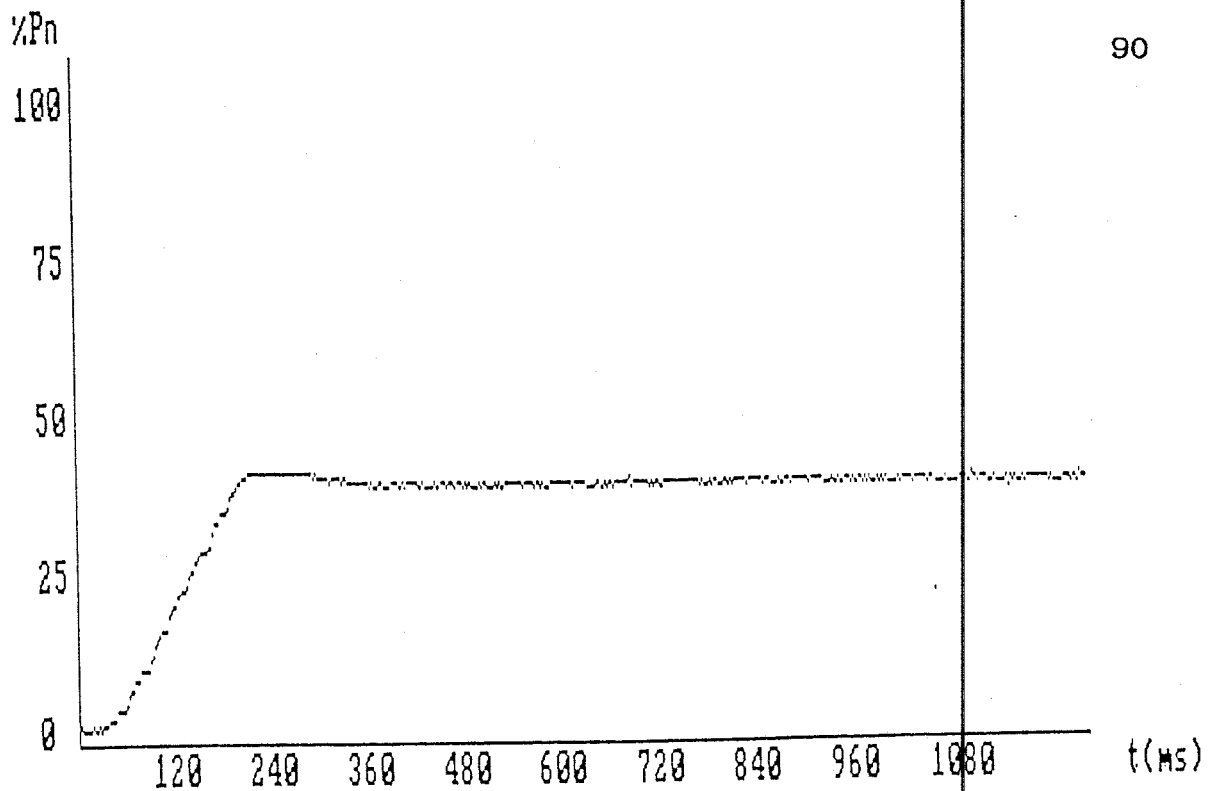
Gambar 4.3. Grafik respon motor untuk K_p , K_d , K_i , ST , berturut-turut 1, 0, 15, 75%



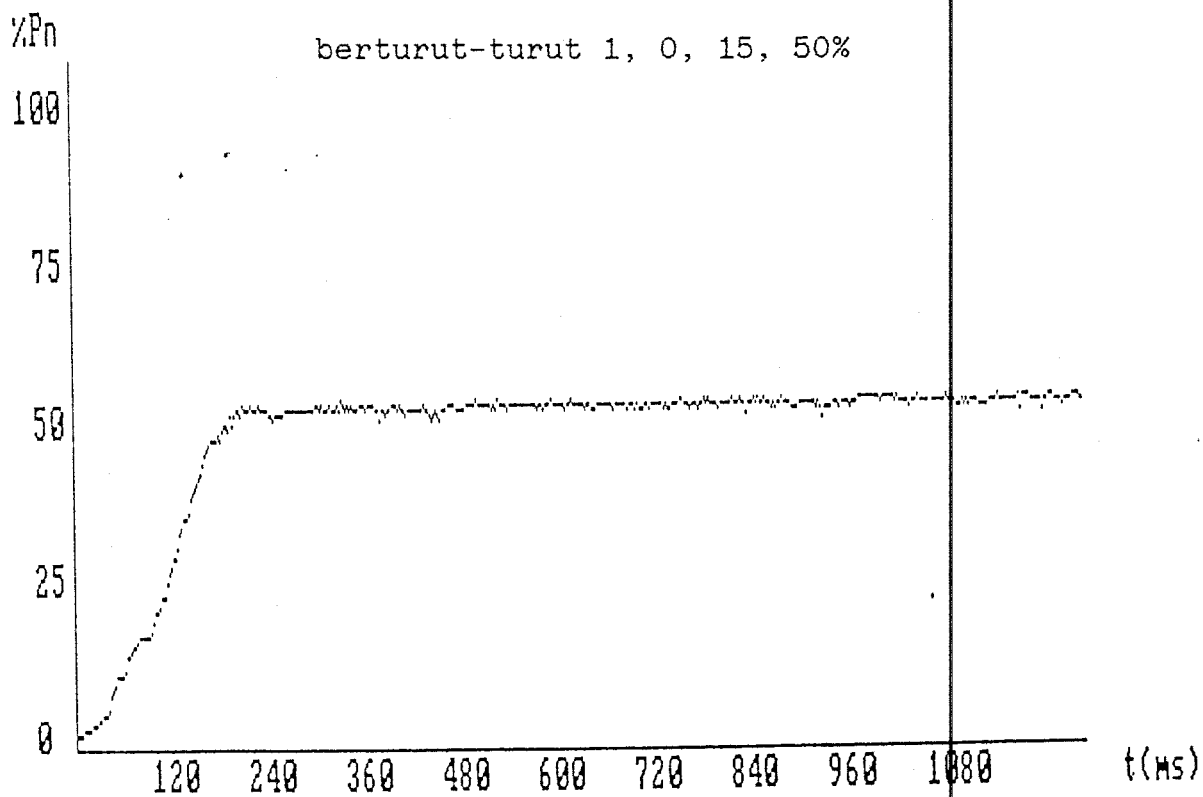
Gambar 4.4. Grafik respon motor untuk K_p , K_d , K_i , ST , berturut-turut 1, 0, 0, 50%



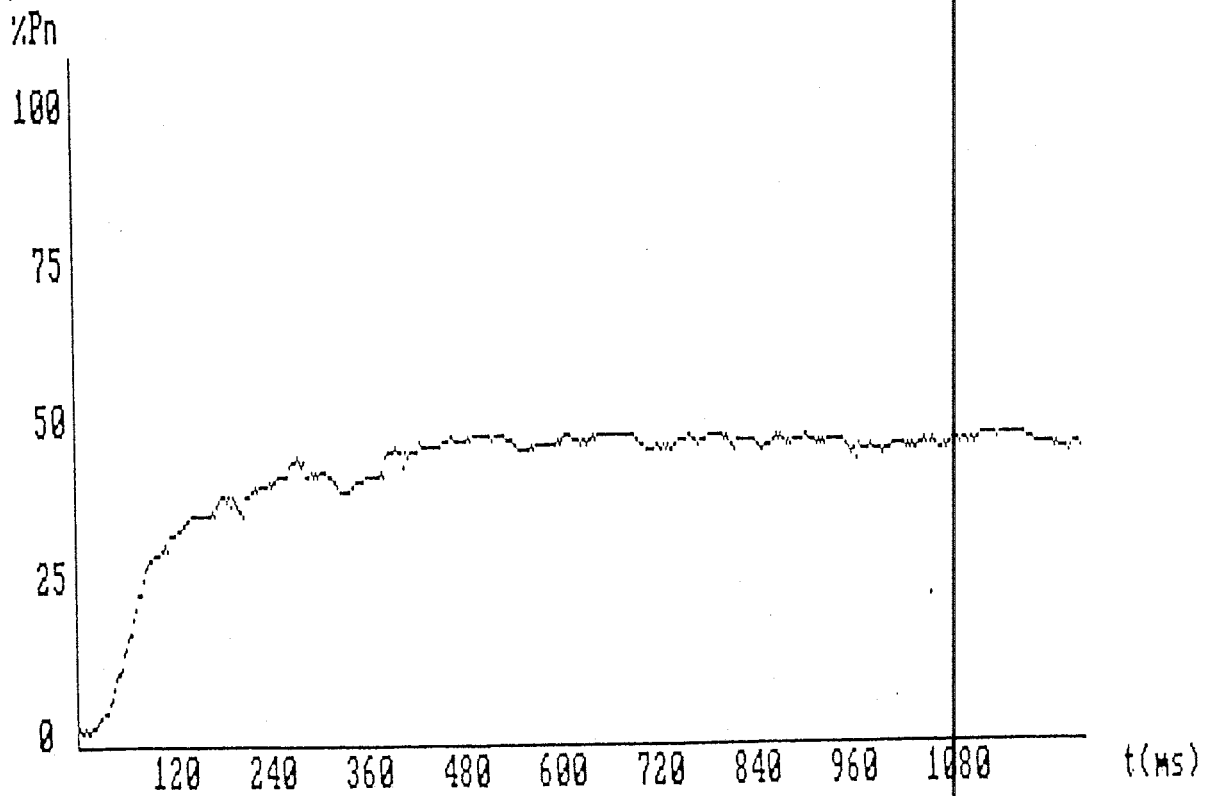
Gambar 4.5. Grafik respon motor untuk K_p , K_d , K_i , ST , berturut-turut 1, 0, 1, 50%



Gambar 4.6. Grafik respon motor untuk K_p , K_d , K_i , ST , berturut-turut 1, 0, 15, 50%



Gambar 4.7. Grafik respon motor untuk K_p , K_d , K_i , ST , berturut-turut 1, 0.03, 1, 50%



Gambar 4.8. Grafik respon motor untuk Kp, Kd, Ki, ST,
berturut-turut 1, 0, 5, 50%

BAB V

KESIMPULAN

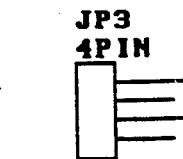
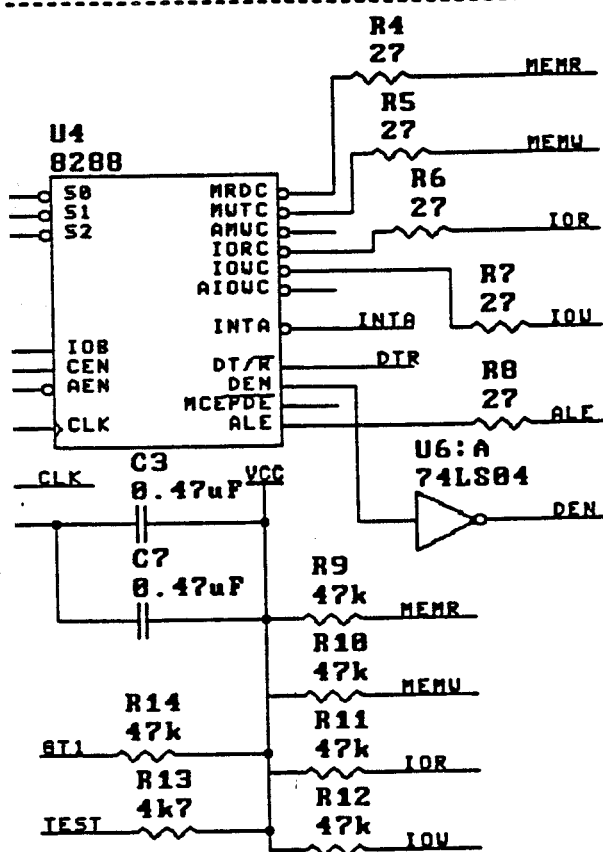
Dari keseluruhan perencanaan dan pembuatan alat diambil beberapa kesimpulan sebagai berikut :

1. Secara teoritis konfigurasi mode maksimum sistem μP 8088 dengan Math Coprocessor 8087 memberikan kelebihan-kelebihan dalam kecepatan perhitungan aritmatika serta ketelitian data yang diperoleh. Karenanya Math Coprocessor yang bekerja paralel dengan μP 8088 sangat cocok dipergunakan untuk proses-proses perhitungan dengan kecepatan tinggi.
2. Untuk menerapkan μP 8088 dalam Mode Maksimum dibutuhkan adanya suatu rangkaian pengontrol bus untuk memberikan pengaturan terhadap operasi bus cycle yang meliputi memory maupun I/O read/write, interrupt, maupun proses lain pada sistem prosessor. Karena semua sinyal-sinyal kontrol pada μP 8088 telah berganti fungsi maupun karakteristiknya.
3. Untuk memberikan cukup waktu bagi suatu device yang lambat, diperlukan suatu rangkaian wait state untuk memperpanjang bus cycle μP 8088.
4. Kontroller PID digital merupakan alternatif penggunaan kontroller PID analog dengan beberapa keunggulan yang dimiliki antara lain : mempunyai fleksibilitas tinggi karena koefisien-koefisien dapat dapat diubah-ubah secara software.

DAFTAR PUSTAKA

1. Astrom, Karl J. and Wittenmark, Bjorn, COMPUTER - CONTROLLED SYSTEMS THEORY AND DESIGN, 2nd Edition, Prentice-Hall International Editions, 1990.
2. Brey, Barry B., THE INTEL MICROPROCESSORS: 8086/8088, 80186, 80286, 80386, AND 80486 ARCHITECTURE, PROGRAMMING AND INTERFACING, 2nd Edition.
3. Coughlin, Robert F. and Driscoll Frederick, OPERATIONAL AMPLIFIERS AND LINIER INTEGRATED CIRCUITS, Prentice-Hall Inc, New Jersey, 1982.
4. Goldsbough, Paul F. and Rony, Peter R., MICROCOMPUTER INTERFACING WITH THE 8255 PPI CHIP.
5. Hall, Douglas V., MICROPROCESSOR AND INTERFACING, PROGRAMMING AND HARDWARE, McGraw-Hill Book Company, 1986.
6. Hogenboom, P., DATA SHEET BOOK 3 CATATAN APLIKASI, PT Elex Media Komputindo Kelompok Gramedia - Jakarta.
7. Jacob, J. Michael, INDUSTRIAL CONTROL ELECTRONICS, APPLICATION AND DESIGN, Prentice-Hall International, Inc.
8. Partoharsodjo, Hartono, TUNTUNAN PRAKTIS PEMROGRAMAN BAHASA ASSEMBLY, PT Elex Media Komputindo Kelompok Gramedia-Jakarta.
9. Steeman, J. P. M., DATA SHEET BOOK 2, PT Elex Media Komputindo Kelompok Gramedia-Jakarta, 1988

10. Uffenbeck, John, THE 8086 /8088 FAMILY, DESIGN
PROGRAMING AND INTERFACING.

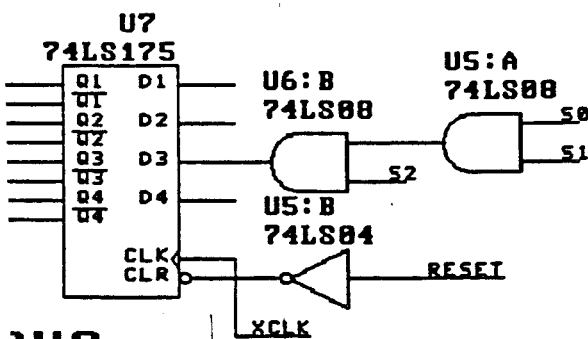


DECODE

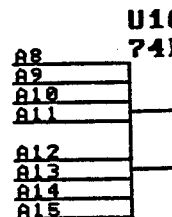
A17
A18
A19



I/O



BUS



05 MAY 1993

EE.1799 TUGAS AKHIR - 6 SKS

Nama Mahasiswa : I G.A.K.Darma Wirata
Nomor Pokok : 2882201029
Bidang Studi : Elektronika
Tugas Diberikan : April
Tugas Diselesaikan : Oktober
Judul Tugas Akhir : Penerapan Math Coprocessor 8087 Pada
Maksimum Sistem 8088 Sebagai Sarana
Praktikum.
Dosen Pembimbing : I. Ir.Karyadi,MSc.
II.Ir.Harris Pirngadi
Dosen Wali : Ir.Soetikno

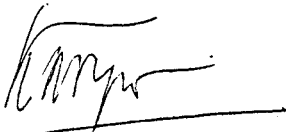
Uraian Tugas Akhir:

Operasi perhitungan dengan mikroprosesor 8088 dapat dilakukan dengan kecepatan dan keakuratan yang tinggi. Namun memiliki keterbatasan. Terutama pada proses memakai bilangan real. Yaitu tingkat ketelitian, proses yang lama, serta besar atau kecil bilangan yang diproses. Untuk mengatasi dibutuhkan prosesor tambahan yaitu math coprocessor 8087.

Pada tugas akhir ini akan dirancang suatu modul yang mengaplikasikan math coprocessor 8087 pada mode maksimum 8088. Dengan maksud sebagai sarana praktikum memprogram dan menerapkan 8087 untuk aplikasi yang sederhana. Sehingga dapat dipergunakan di laboratorium bagi mahasiswa yang ingin mempelajari 8087.

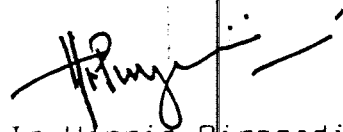
Surabaya, April 1993

Dosen Pembimbing I



(Ir.Karyadi,MSc.)
NIP. 130 520 754


Dosen Pembimbing II



(Ir.Harris Pirngadi)
NIP. 131 843 903

Mengetahui

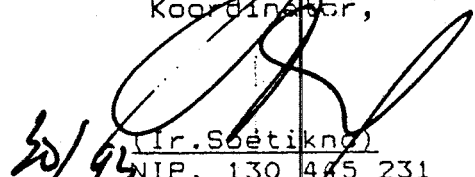
Jurusan Teknik Elektro
Ketua,



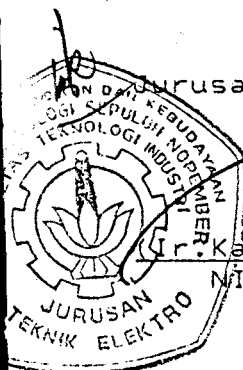
(Ir.Katjuk Astrowulan,MSEE)
NIP. 130 687 438

Menyetujui

Bidang Studi Elektronika
Koordinator,



(Ir.Soetikno)
NIP. 130 445 231



A. JUDUL : PENERAPAN MATH COPROCESSOR 8087 PADA MAKSIMUM SISTEM 8088 SEBAGAI SARANA PRAKTIKUM.

B. RUANG LINGKUP : - Elektronika
- Mikroelektronika
- Mikroprosesor
- Instrumentasi Elektronika

C. LATAR BELAKANG : Kemampuan mikroprosesor 8088 dalam melakukan operasi aritmatika sangat andal. Terutama dalam hal kecepatan dan ketepatannya. Namun dalam operasi dengan bilangan real terdapat kendala. Yaitu tingkat ketelitian, proses yang lebih lama, serta orde bilangan yang mampu dioperasikan. Untuk operasi dengan bilangan real diperlukan prosesor pembantu yaitu math coprocessor 8087. Pada tugas akhir ini akan dirancang sarana yang mengaplikasikan coprocessor ini pada mode maksimum 8088.

D. PENELAAHAN STUDI: Secara prinsip ada empat bagian yang membentuk alat ini. Pertama unit mikroprosesor sendiri sebagai pemroses utama, unit memori sebagai penyimpan program atau data, unit keyboard untuk data input, dan unit tampilan. Untuk interface peralatan eksternal disediakan slot tersendiri. Mikroprosesor memerlukan penunjang berupa rangkaian pewaktu, rangkaian reset, rangkaian demultiplek alamat dan data. Coprocessor karena sebagai prosesor tambahan, terhubung paralel dengan mikroprosesor 8088. Untuk pengaturan sinyal-sinyal baca atau tulis dari atau ke memori atau port

diperlukan pengontrol bus yang terpisah.

E. TUJUAN : Merencanakan dan membuat modul maksimum sistem 8088 dengan menerapkan coprocessor 8087 untuk sarana praktikum.

F. LANGKAH-LANGKAH: 1. Studi Literatur
2. Perencanaan Alat
3. Pembuatan Alat
4. Pengujian dan Pengukuran
5. Penulisan Laporan

G. JADWAL KEGIATAN :

BULAN KE

KEGIATAN	I	II	III	IV	V	VI
1						
2						
3						
4						
5						

H. RELEVANSI : Diharapkan dapat dipakai pada laboratorium-laboratorium sebagai sarana praktikum penerapan math coprocessor 8087.

RIWAYAT HIDUP



I G.A.K. Darma Wirata dilahirkan di Denpasar pada tanggal 30 April 1969. Putra keempat dari Bapak IG.M. Putra dan Ibu IG.A. Rai Suryadi yang bertempat tinggal di Jl. Ngurah Rai 24 Singaraja.

Terdaftar sebagai mahasiswa Institut Teknologi Sepuluh Nopember pada tahun 1988 dengan nomor registrasi pokok 2882201029.

Pendidikan yang telah ditempuh sampai saat ini :

- SD Mutiara Singaraja, lulus tahun 1982
- SMP Mutiara Singaraja, lulus tahun 1985
- SMA Negeri 1 Singaraja, lulus tahun 1988

Dan diharapkan lulus pada ujian sarjana Fakultas Teknologi Industri Jurusan Teknik Elektro ITS pada periode Agustus 1994.